

ENUMERATION AND RANDOM GENERATION OF
UNLABELED CLASSES OF GRAPHS:
A PRACTICAL STUDY OF CYCLE POINTING
AND THE DISSYMMETRY THEOREM

Alexander Daniel Iriza

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering

Princeton University
Department of Computer Science

Advised by Jérémie Lumbroso and Robert Sedgewick

September 2015



Contents

1	Introduction	4
2	Analysis of unrooted graph classes	5
2.1	Enumeration	5
2.2	Boltzmann sampling	6
2.3	The challenges of unrooted graphs	7
2.3.1	The labeled case	7
2.3.2	The unlabeled case	10
3	The dissymmetry theorem	11
3.1	Overview	11
3.2	Boltzmann sampler by center-rejection	12
3.3	Example	14
4	Cycle pointing	16
4.1	Introduction	17
4.2	Decomposition of cycle-pointed classes	22
4.2.1	Theory	22
4.2.2	Example	24
4.3	Enumeration	26
4.3.1	Theory	26
4.3.2	Example	28
4.4	Boltzmann samplers	30
4.4.1	Theory	30
4.4.2	Example	32
5	Two interesting classes of graphs	33
5.1	Distance-hereditary graphs	33
5.2	Three-leaf power graphs	33
5.3	Split decomposition	33
5.4	Enumeration with the dissymmetry theorem	37
5.4.1	Distance-hereditary graphs	37
5.4.2	Three-leaf power graphs	38
5.5	Enumeration and sampling with cycle pointing	39
5.5.1	Distance-hereditary graphs	39
5.5.2	Three-leaf power graphs	41
6	Implementation and empirical study of samplers	43
6.1	Overview	43
6.2	Details of interest	46
6.2.1	Oracles	46
6.2.2	Radius of convergence	46
6.2.3	Sampling of random variables	47
6.3	Empirical analysis	48
6.3.1	Chi-squared tests	48
6.3.2	Timing studies	49
7	Conclusion	50
A	Commonly used notation	52

B Unbiased sampler for the class of unlabeled, unrooted, non-plane 2-3 trees	53
References	58

Abstract

Our work studies the enumeration and random generation of unlabeled combinatorial classes of unrooted graphs. While the technique of vertex pointing provides a straightforward procedure for analyzing a labeled class of unrooted graphs by first studying its rooted counterpart, the existence of nontrivial symmetries in the unlabeled case causes this technique to break down. Instead, techniques such as the dissymmetry theorem (of Otter [31]) and cycle pointing (of Bodirsky *et al.* [18, 19]) have emerged in the unlabeled case, with the former providing an enumeration of the class and the latter providing both an enumeration and an unbiased sampler. In this work, we extend the power of the dissymmetry theorem by showing that it in fact provides a Boltzmann sampler for the class in question. We then present an exposition of the cycle pointing technique, with a focus on the enumeration and random generation of the underlying unpointed class. Finally, we apply cycle pointing to enumerate and implement samplers for the classes of distance-hereditary graphs and three-leaf power graphs.

1 Introduction

The study of families of graphs – enumerating them, randomly generating them, and analyzing their parameters and asymptotics – has been one of the many great success stories of analytic combinatorics (see, for example, the discussion by Flajolet and Sedgewick [28, Sec. II.5.]). These graphs may come in many varieties – labeled or unlabeled, rooted or unrooted, plane or non-plane, and with or without cycles – and the techniques for analyzing these variations are just as numerous.

The typical first step in the analysis is to build a combinatorial specification for the class in question. In the case of graphs this specification is usually recursive, expressing a graph as a combination of smaller elements of the same class or related classes. If the class happens to be rooted, then the root of a graph, whether a vertex, edge, or other feature, provides a distinguished location at which it can be decomposed and by which a specification of the class can be written. However if the graphs in the class are unrooted, no such distinguished location exists.

To address this difficulty, techniques have arisen in both the labeled and unlabeled cases which analyze an unrooted class by relating it to different rooted (also sometimes called *marked* or *pointed*) versions of itself. If the class is labeled, meaning that the vertices of each graph of size n are given distinct labels from 1 to n , then there are n distinct vertex-rooted graphs for each unrooted graph of size n , and this fact allows for an analysis of the vertex-rooted class to be easily translated into an analysis of the unrooted class [28], and even a random sampler [4, §2.2.1]. However if the class is unlabeled, meaning that the vertices of a graph are distinguished not by distinct labels but rather only by their adjacencies to other vertices in the graph, then two different graphs of the same size may have two different sets of non-trivial symmetries (reflections, rotations, etc.) and therefore two different numbers of rooted graphs corresponding to them.

In order then to analyze an unlabeled, unrooted class of graphs, a common strategy is to study not only the corresponding vertex-rooted class but also to study corresponding classes that are rooted at features other than vertices. One example of such a strategy is the dissymmetry theorem, introduced by Otter in 1948 [31] and popularized by Bergeron *et al.* in 1997 [10], which allows one to compute the enumeration of various unlabeled, unrooted classes of *trees* by enumerating the corresponding classes that are rooted at a vertex, an edge, and a directed edge. A second example is the technique of cycle pointing, introduced by Bodirsky *et al.* [18, 19], in which a graph is rooted at a cycle on some of its vertices satisfying a certain property, and this property is chosen so that there are n distinct cycle-pointed graphs corresponding to each unrooted graph of size n . Cycle pointing allows one both to compute the enumerations of unlabeled, unrooted classes of graphs, and to construct unbiased random generators for these classes.

This work has three main contributions. In Section 3, we introduce a new Boltzmann sampler for an arbitrary class of trees that is specified by the dissymmetry theorem, assuming that there exist samplers for the corresponding vertex-rooted and edge-rooted classes. This allows the dissymmetry

theorem to be used not only for enumeration but also for sampling. The sampler relies on the concept of the *center* of a tree – informally, the vertex or edge of the tree that is farthest from its leaves – and it works by repeatedly drawing a tree from the union of the vertex-rooted and edge-rooted classes until the drawn tree is rooted at its center.

In Section 4 we provide an exposition of the cycle pointing technique, complete with diagrams to visually convey the important concepts. We tailor our exposition to our aims of enumerating and sampling from unlabeled, unrooted classes of graphs, and omit certain aspects of the theory for the sake of clarity.

In Sections 5 and 6, we apply the cycle pointing technique to analyze two unlabeled classes of graphs – distance-hereditary graphs and three-leaf power graphs. Using cycle pointing, we compute exact enumerations for these classes that agree with the ones developed with the dissymmetry theorem by Chauve *et al.* [6]. We then build unbiased samplers for these two classes of graphs using cycle pointing. A full implementation in Maple is provided, along with a description of some of its features, empirical results, and drawings of randomly generated graphs.

2 Analysis of unrooted graph classes

2.1 Enumeration

When studying a combinatorial class \mathcal{A} , one of the first and most fundamental challenges to address is to determine how many objects of a given size exist in the class. The sequence

$$\mathcal{A}_n = \#\{\gamma \in \mathcal{A} \mid |\gamma| = n\}$$

that answers this question is called the *enumeration* of \mathcal{A} , and the formal power series

$$\mathcal{A}(z) = \sum_{n=0}^{\infty} \mathcal{A}_n z^n \quad \mathcal{A}(z) = \sum_{n=0}^{\infty} \frac{\mathcal{A}_n}{n!} z^n$$

are called, respectively, the *ordinary generating function (OGF)* and *exponential generating function (EGF)* of \mathcal{A} . The former is used in the case when \mathcal{A} is unlabeled, and the latter in the case when \mathcal{A} is labeled, so no confusion should arise from this overloaded notation.

If \mathcal{A} is *decomposable*, meaning that it can be specified recursively in terms of basic classes (ε , \mathcal{Z} , SET, SEQ, CYC, etc.), itself, other decomposable classes, and operators (disjoint union, product, substitution, etc.), then by the theory of symbolic transfer theorems [28] its combinatorial specification immediately gives a generating function equation that can often be solved in order to recover the coefficients \mathcal{A}_n .

For example, let \mathcal{C} be the class of Cayley trees, which are labeled, rooted, non-plane trees. An element $\gamma \in \mathcal{C}$ consists of a root connected to a set of 0 or more elements of \mathcal{C} , so we have the recursive specification

$$\mathcal{C} = \mathcal{Z} \times \text{SET}(\mathcal{C})$$

for the class. This results in the exponential generating function equation

$$\mathcal{C}(z) = z \cdot \exp(\mathcal{C}(z)),$$

and by the Lagrange Inversion theorem [28] it follows that

$$\mathcal{C}_n = n! \cdot \frac{1}{n} [u^{n-1}] e^{nu} = (n-1)! \frac{n^{n-1}}{(n-1)!} = n^{n-1}.$$

When \mathcal{A} is not decomposable, however, this method is not sufficient for computing its enumeration, because its key tool – a symbolic specification for \mathcal{A} – is missing. In this case a variety of other techniques may apply, depending on the particular nature of \mathcal{A} , and in this work we will focus on the techniques used for one important family of non-decomposable combinatorial classes: classes of unrooted graphs.

2.2 Boltzmann sampling

After studying the enumeration of a class, a natural next step is to investigate potential methods of randomly generating objects from this class. This can be useful in order to visualize large random objects in the class, and to study the behavior of parameters of the objects as their size grows.

One of the first proposed methods for randomly generating objects from a combinatorial class was the *recursive method* of Flajolet *et al.* [30], which uses the enumeration of the class (*i.e.* the coefficients of its generating function) to sample an object of a specified size uniformly at random. More recently, *Boltzmann samplers* have been introduced by Duchon *et al.* as a general technique to sample objects from an arbitrary *decomposable* combinatorial class [27, 29, 18]. Indeed, the rules outlined in these articles allow for an automatic, algorithmic translation of the combinatorial specification of the class into a Boltzmann sampler for that class. Boltzmann samplers are particularly attractive because they are more efficient than the recursive method, running in linear time in the size of the output and not requiring the linear-time precomputation of the recursive method, and they rely not on the individual terms of the enumeration but instead on basic constructs from probability theory and the ability to evaluate the generating function of the class.

Definition 2.1. Let \mathcal{A} be an unlabeled combinatorial class, and let $\mathcal{A}(z)$ be its OGF. For a fixed parameter value $z > 0$ at which $\mathcal{A}(z)$ converges, an *ordinary Boltzmann sampler* $\Gamma\mathcal{A}(z)$ is a random generator that draws an object $\gamma \in \mathcal{A}$ with probability

$$\mathbb{P}_z[\gamma] = \frac{z^{|\gamma|}}{\mathcal{A}(z)}.$$

Since the only property of γ upon which $\mathbb{P}_z[\gamma]$ depends is its size, we see that a Boltzmann sampler is *unbiased*, in the sense that it draws all objects of a given size in \mathcal{A} with equal probability. However unlike in recursive sampling, it is not possible to specify at the outset the size of the object that will be returned. Instead, this size is a random variable S whose distribution depends on the parameter z as follows:

$$\begin{aligned} \mathbb{P}_z[S = n] &= \sum_{\substack{\gamma \in \mathcal{A}, \\ |\gamma| = n}} \frac{z^{|\gamma|}}{\mathcal{A}(z)} = \frac{\mathcal{A}_n z^n}{\mathcal{A}(z)}, \\ \mathbb{E}_z[S] &= \sum_{n=0}^{\infty} \frac{n \mathcal{A}_n z^n}{\mathcal{A}(z)} = \frac{z \mathcal{A}'(z)}{\mathcal{A}(z)}. \end{aligned}$$

A parallel definition holds when \mathcal{A} is a labeled class, except that $\mathcal{A}(z)$ must be the EGF of the class and the expressions for $\mathbb{P}_z[\gamma]$ and $\mathbb{P}_z[S = n]$ must be scaled by $1/n!$.

Duchon *et al.* [27] provide a set of rules for automatically building a Boltzmann sampler for an arbitrary labeled decomposable class, and this theory was extended to unlabeled classes by Flajolet *et al.* [29] and Bodirsky *et al.* [18]. Some of the basic rules in the labeled case are shown in Table 1, where the Poisson, geometric, and logarithmic distributions are the power series distributions for the functions e^z , $1/(1-z)$, and $\log(1/(1-z))$, respectively. The first four rules apply in the unlabeled case as well, while the rules for unlabeled sets, sequences, and cycles will be discussed in Section 4.4.1.

To see a concrete example (which we will return to in Section 2.3.1), we consider the class \mathcal{C} of Cayley trees. This class can be specified by

$$\mathcal{C} = \mathbb{Z} \times \text{SET}(\mathcal{C}),$$

and by the rules above a Boltzmann sampler for this class is given by:

$$\Gamma\mathcal{C}(z) = (T \leftarrow (v = \bullet; \underbrace{\Gamma\mathcal{C}(z), \dots, \Gamma\mathcal{C}(z)}_{\text{Pois}(\mathcal{C}(z))}); \text{ return } (\text{label}(T), v));$$

Class	Boltzmann sampler
$\mathcal{C} = \varepsilon$	$\Gamma\mathcal{C}(z) = \circ$ (atom of size 0)
$\mathcal{C} = \mathcal{Z}$	$\Gamma\mathcal{C}(z) = \bullet$ (atom of size 1)
$\mathcal{C} = \mathcal{A} + \mathcal{B}$	$\Gamma\mathcal{C}(z) = \text{if Bern}\left(\frac{\mathcal{A}(z)}{\mathcal{A}(z) + \mathcal{B}(z)}\right) \text{ then } \Gamma\mathcal{A}(z) \text{ else } \Gamma\mathcal{B}(z)$
$\mathcal{C} = \mathcal{A} \times \mathcal{B}$	$\Gamma\mathcal{C}(z) = (\Gamma\mathcal{A}(z), \Gamma\mathcal{B}(z))$
$\mathcal{C} = \text{SET}(\mathcal{A})$	$\Gamma\mathcal{C}(z) = \underbrace{(\Gamma\mathcal{A}(z), \dots, \Gamma\mathcal{A}(z))}_{\text{Pois}(\mathcal{A}(z))}$
$\mathcal{C} = \text{SEQ}(\mathcal{A})$	$\Gamma\mathcal{C}(z) = \underbrace{(\Gamma\mathcal{A}(z), \dots, \Gamma\mathcal{A}(z))}_{\text{Geom}(\mathcal{A}(z))}$
$\mathcal{C} = \text{CYC}(\mathcal{A})$	$\Gamma\mathcal{C}(z) = \underbrace{(\Gamma\mathcal{A}(z), \dots, \Gamma\mathcal{A}(z))}_{\text{Loga}(\mathcal{A}(z))}$

Table 1. Boltzmann sampler rules for labeled classes.

where the first element v of the tuple is an atom denoting the root of the tree, the subtrees that appear after the semicolon in the tuple are its children, and the *label* function assigns a random permutation of the labels $1, 2, \dots, |T|$ to the atoms of T .

As is the case with enumeration, this technique does not provide Boltzmann samplers for classes that are not decomposable, because it relies on a recursive specification for the class in terms of classes whose Boltzmann samplers have already been constructed.

2.3 The challenges of unrooted graphs

Our aim in this work is to study techniques for enumerating and sampling from various classes of unrooted graphs. One issue that arises here is that such graphs have no “distinguished” vertex, edge, or other feature at which they can be recursively decomposed into smaller elements of the same class or other classes; instead, they are simply a set of vertices, together with a set of edges connecting certain pairs of those vertices. Thus, techniques beyond the ones described in Sections 2.1 and 2.2 are needed. We now introduce some of these techniques, first in the case when the class is labeled, and then in the more challenging case when it is unlabeled.

2.3.1 The labeled case

In order to study a class of labeled, unrooted graphs, a useful technique is to begin by studying the corresponding class of graphs that are rooted at a vertex. This is known as *vertex-rooting* or *vertex-pointing*, and the intuition behind it is straightforward: for any labeled, unrooted graph with n nodes, there are exactly n vertex-rooted graphs corresponding to it (since the root can be chosen as the vertex labeled 1, the vertex labeled 2, \dots , or the vertex labeled n), so there is a 1-to- n correspondence between the size- n elements of the unrooted class and the size- n elements of the rooted class.

If the rooted class is decomposable (as was the class of Cayley trees in Section 2.1, for instance), the standard techniques can be employed to develop an enumeration and Boltzmann sampler for it. As we will see below, this enumeration and sampler, together with the 1-to- n correspondence, can

be used to enumerate and sample from the unrooted class.

Definition 2.2. For a class \mathcal{A} of unrooted objects, the *vertex-rooted* class corresponding to \mathcal{A} is the class \mathcal{A}^\bullet defined by

$$\mathcal{A}^\bullet = \{(\gamma, v) \mid \gamma \in \mathcal{A} \text{ and } v \text{ is a node of } \gamma\},$$

where the size of an element (γ, v) in \mathcal{A}^\bullet is defined as the size of γ in \mathcal{A} .

Lemma 2.3. For a class \mathcal{A} of labeled, unrooted objects with EGF $\mathcal{A}(z)$, the EGF for \mathcal{A}^\bullet is given by

$$\mathcal{A}^\bullet(z) = z\mathcal{A}'(z).$$

Proof. There are n objects of size n in \mathcal{A}^\bullet for each graph of size n in \mathcal{A} , so $\mathcal{A}_n^\bullet = n\mathcal{A}_n$. Thus the EGF for \mathcal{A}^\bullet is

$$\mathcal{A}^\bullet(z) = \sum_{n=0}^{\infty} \frac{\mathcal{A}_n^\bullet}{n!} z^n = \sum_{n=0}^{\infty} \frac{n\mathcal{A}_n}{n!} z^n = z \sum_{n=0}^{\infty} \frac{\mathcal{A}_n}{n!} n z^{n-1} = z\mathcal{A}'(z).$$

□

Lemma 2.4. For a class \mathcal{A} of labeled objects,

$$\mathcal{A}_n = \frac{1}{n} \mathcal{A}_n^\bullet.$$

Furthermore, if $\Gamma\mathcal{A}^\bullet(z)$ is a Boltzmann sampler for \mathcal{A}^\bullet , then

$$\tilde{\Gamma}\mathcal{A}(z) = \{(\gamma, v) \leftarrow \Gamma\mathcal{A}^\bullet(z); \text{ return } \gamma; \}$$

is an unbiased sampler for \mathcal{A} , in the sense that

$$\mathbb{P}_z[\gamma \mid |\gamma| = n] = \frac{1}{\mathcal{A}_n}.$$

Proof. By the discussion in the first paragraph of this section, we see that \mathcal{A}_n is the correct enumeration for \mathcal{A} . For the sampler, the probability of drawing γ from $\tilde{\Gamma}\mathcal{A}(z)$ is equal to the probability of drawing (γ, v) from $\Gamma\mathcal{A}^\bullet(z)$ for some vertex v of γ , and since $\Gamma\mathcal{A}^\bullet(z)$ is a Boltzmann sampler for \mathcal{A}^\bullet , this probability is

$$\mathbb{P}_z[\gamma] = \sum_{v \text{ is a vertex of } \gamma} \frac{z^{|\gamma|}}{|\gamma|! \mathcal{A}^\bullet(z)} = \frac{|\gamma| z^{|\gamma|}}{|\gamma|! z \mathcal{A}'(z)} = \frac{z^{|\gamma|-1}}{(|\gamma|-1)! \mathcal{A}'(z)}.$$

Since the only property of γ on which this expression depends is $|\gamma|$, it follows that $\tilde{\Gamma}\mathcal{A}(z)$ draws all objects of a given size from \mathcal{A} with equal probability, and hence is unbiased. □

We note, however, that $\mathbb{P}_z[\gamma]$ is not equal to

$$\frac{z^{|\gamma|}}{|\gamma|! \mathcal{A}(z)},$$

so $\tilde{\Gamma}\mathcal{A}(z)$ is not in fact a Boltzmann sampler for \mathcal{A} . In order to obtain a Boltzmann sampler, the technique of rejection may be employed as follows:

Lemma 2.5. The following is a Boltzmann sampler for \mathcal{A} :

$$\Gamma\mathcal{A}(z) = \{\text{do } (\gamma, v) \leftarrow \Gamma\mathcal{A}^\bullet(z) \text{ while } \text{label}(v) \neq 1; \text{ return } \gamma; \}$$

This simple rejection solution is one that has been suggested before, for instance by Bousquet-Mélou and Weller [23, §11.1] to draw random minor-closed classes of graphs. It has been improved upon by Darrasse *et al.* [4, §2.2.1], who, instead of fixing the parameter z , draw it according to a certain differentiated probability distribution which biases the exponential Boltzmann sampler in order to mimic an unrooted distribution. While their technique, which avoids rejection altogether, is suitable for labeled objects, it is unclear how to apply it to the unlabeled objects we will study beginning in Section 2.3.2. Indeed it seems that it does not address how to obtain an enumeration of the unrooted class, but instead assumes that such an enumeration is available (of course, by Lemma 2.4, this enumeration is trivially available in the labeled case).

Proof. The probability of drawing γ from $\Gamma\mathcal{A}(z)$ is equal to the probability of drawing (γ, v) from $\Gamma\mathcal{A}^\bullet(z)$ conditioned on the event that $\text{label}(v) = 1$, which is

$$\begin{aligned} \frac{z^{|\gamma|}}{|\gamma|! \mathcal{A}^\bullet(z)} &= \frac{z^{|\gamma|}}{|\gamma|! \mathcal{A}^\bullet(z)} \cdot \frac{1}{\sum_{n=1}^{\infty} \mathbb{P}_z^{\mathcal{A}^\bullet}[\text{label}(v) = 1 \mid |\gamma| = n] \cdot \mathbb{P}_z^{\mathcal{A}^\bullet}[|\gamma| = n]} \\ &= \frac{z^{|\gamma|}}{|\gamma|! \mathcal{A}^\bullet(z)} \cdot \frac{1}{\sum_{n=1}^{\infty} \frac{1}{n} \frac{\mathcal{A}_n^\bullet z^n}{n! \mathcal{A}^\bullet(z)}} \\ &= \frac{z^{|\gamma|}}{|\gamma|! \sum_{n=1}^{\infty} \frac{1}{n} \frac{n \mathcal{A}_n}{n!} z^n} \\ &= \frac{z^{|\gamma|}}{|\gamma|! \mathcal{A}(z)}. \end{aligned}$$

□

To see an example of this in action, let \mathcal{T} be the class of labeled non-plane trees, which are connected graphs with no cycles. Then \mathcal{T}^\bullet is the class of labeled rooted non-plane trees, *i.e.* Cayley trees, whose enumeration is given (from Section 2.1) by

$$\mathcal{T}_n^\bullet = n^{n-1}.$$

Thus the number of labeled trees with n vertices is

$$\mathcal{T}_n = \frac{1}{n} \mathcal{T}_n^\bullet = n^{n-2}.$$

Furthermore, the class \mathcal{T}^\bullet has a Boltzmann sampler

$$\Gamma\mathcal{T}^\bullet(z) = \{T \leftarrow (v; \underbrace{\Gamma\mathcal{T}^\bullet(z), \dots, \Gamma\mathcal{T}^\bullet(z)}_{\text{Pois}(\mathcal{T}^\bullet(z))}); \text{ return } (\text{label}(T), v); \}$$

so a Boltzmann sampler for \mathcal{T} is given by

$$\Gamma\mathcal{T}(z) = \{\text{do } (T, v) \leftarrow \Gamma\mathcal{T}^\bullet(z) \text{ while } \text{label}(v) \neq 1; \text{ return } T; \}.$$

2.3.2 The unlabeled case

The analysis of unlabeled, unrooted classes of graphs poses a greater challenge than the analysis of their labeled counterparts, and the techniques described in the previous section do not suffice in general. The difficulty here arises from the existence of symmetries: without labels on the vertices, a graph may have internal symmetries that cause some of its vertices to be indistinguishable from each other, and rooting at two indistinguishable vertices will give rise to identical rooted graphs. These symmetries are of course not the same for all graphs of a given class and size. Thus while it is still possible to build the vertex-rooted class for a given unrooted class, it is no longer the case that each unrooted graph of size n gives rise to the same number of rooted graphs of size n .

For example, consider the two graphs of size 4 in Figure 1. If the vertices have distinct labels,

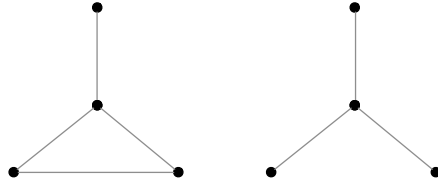


Figure 1. Two graphs of size 4.

then each gives rise to 4 distinct rooted graphs, one for each vertex. However if the vertices are unlabeled, the first graph gives rise to 3 distinct rooted graphs while the second gives rise to only 2, as seen in Figure 2.

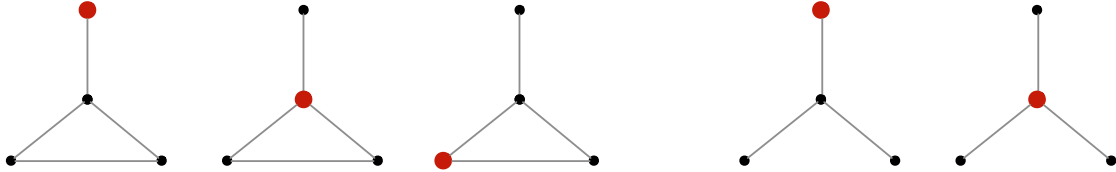


Figure 2. Rooted versions of the graphs in Figure 1.

Without the 1-to- n correspondence between the unrooted and rooted classes, the techniques from the previous section cannot be directly applied to derive the enumeration and Boltzmann sampler of the unrooted class from those of the rooted class. In this work we study two techniques for addressing this difficulty. In Section 3 we discuss the dissymmetry theorem, which combines the techniques of pointing at vertices, undirected edges, and directed edges in order to develop enumerations and Boltzmann samplers for unlabeled, unrooted classes of graphs. Then in Section 4 we discuss cycle pointing, a technique which points at certain cycles of vertices of a graph in such a way that each unpointed object of size n gives rise to exactly n pointed objects of size n . This establishes a 1-to- n correspondence for unlabeled graphs that can be employed in much the same way that vertex pointing was used in the labeled case.

3 The dissymmetry theorem

3.1 Overview

The dissymmetry theorem, first introduced by Otter [31] and popularized by Bergeron *et al.* [10] in the context of species theory, relates the enumeration of an unrooted class of trees to the enumerations of three corresponding rooted classes. While it may seem restrictive that this theorem applies only to classes of trees, many non-tree classes of graphs – some of which we will see in Section 5 – can be characterized in terms of structures that are trees. The dissymmetry theorem can then be applied to these classes as well, as mentioned, for instance, by Chapuy *et al.* [12].

Theorem 3.1. (*Dissymmetry theorem*) *Let \mathcal{A} be an unrooted class of trees, and let \mathcal{A}^\bullet , $\mathcal{A}^{\bullet-\bullet}$, and $\mathcal{A}^{\bullet\rightarrow\bullet}$ be the corresponding classes of trees rooted at a vertex, an edge, and a directed edge, respectively. Then there is a bijection*

$$\mathcal{A} + \mathcal{A}^{\bullet\rightarrow\bullet} \simeq \mathcal{A}^\bullet + \mathcal{A}^{\bullet-\bullet},$$

in the sense that for a given n , there are the same number of objects of size n in both classes. In other words,

$$\mathcal{A}_n = \mathcal{A}_n^\bullet + \mathcal{A}_n^{\bullet-\bullet} - \mathcal{A}_n^{\bullet\rightarrow\bullet}.$$

An elegant proof of this result is given by Drmota [22], which we briefly review here.

Definition 3.2. Let T be a tree. We define the *center* of T to be the vertex or edge of T that is obtained by the following iterative procedure: at each step, simultaneously delete all leaves of T , and repeat until T has size less than 3. Since every tree has at least one leaf, and deleting a leaf from a tree results in another tree, this process will terminate with either a single vertex or a single edge of T .

Proof of Theorem 3.1. Consider each tree $T \in \mathcal{A}$ as being rooted at its center, which is either a vertex or an edge of T . Then the class

$$\mathcal{A}' = (\mathcal{A}^\bullet + \mathcal{A}^{\bullet-\bullet}) \setminus \mathcal{A}$$

can be thought of as the subclass of $\mathcal{A}^\bullet + \mathcal{A}^{\bullet-\bullet}$ containing all vertex-rooted and edge-rooted trees that are not rooted at their center.

It suffices to show a bijection between this class and $\mathcal{A}^{\bullet\rightarrow\bullet}$. Consider a rooted tree $(T, r) \in \mathcal{A}'$, with root r and center $c = c(T)$. There are four possible cases to consider, which are outlined below – in each case we define a mapping $\phi((T, r)) = (T, e)$ to a tree $(T, e) \in \mathcal{A}^{\bullet\rightarrow\bullet}$, and afterwards we check that this mapping is indeed a bijection. Also, in each case we denote by P be the unique path from r to c in T (where P contains both endpoints of r (and/or c) if r (and/or c) is an edge).

1. r and c are vertices (note that $r \neq c$ by assumption)

Since $r \neq c$, the length of P is ≥ 1 . Let e be the directed edge obtained by directing the first edge of P away from r .

2. r is a vertex and c is an edge

Let e be the directed edge obtained by directing the first edge of P away from r .

3. r and c are edges (note that $r \neq c$ by assumption)

Let e be the directed edge obtained by directing the first edge of P (namely r) away from c – since $r \neq c$, this is well-defined.

4. r is an edge and c is a vertex

Let e be the directed edge obtained by directing the first edge of P (namely r) away from c .

To show that ϕ is a bijection, it suffices to show that it has an inverse. Indeed, for $(T, e) \in \mathcal{A}^{\bullet \rightarrow \bullet}$ with center c , define $\phi^{-1}((T, e)) = (T, r)$ as follows:

1. $e = c$, or $e \neq c$ and e is directed towards c

Let r be the tail of e .

2. $e \neq c$ and e is directed away from c

Let r be e with its direction removed.

By inspection we see that ϕ and ϕ^{-1} are inverses, so $\phi : \mathcal{A}' \rightarrow \mathcal{A}^{\bullet \rightarrow \bullet}$ is a bijection. \square

Following Chauve *et al.* [6], we note that it is possible to only consider internal nodes when applying the dissymmetry theorem:

Lemma 3.3. *The dissymmetry theorem remains true when the three rooted classes are restricted to only contain those trees rooted at internal nodes or edges between two internal nodes.*

Proof. Let \square and \blacksquare denote a leaf and an internal node, respectively; so, for example, $\mathcal{A}^{\square \rightarrow \blacksquare}$ is the class of trees in \mathcal{A} rooted at a directed edge from a leaf to an internal node. Then

$$\begin{aligned}\mathcal{A}^\bullet &= \mathcal{A}^\square + \mathcal{A}^\blacksquare \\ \mathcal{A}^{\bullet \rightarrow \bullet} &= \mathcal{A}^{\square \rightarrow \square} + \mathcal{A}^{\square \rightarrow \blacksquare} + \mathcal{A}^{\blacksquare \rightarrow \square} + \mathcal{A}^{\blacksquare \rightarrow \blacksquare} \\ \mathcal{A}^{\bullet \rightarrow \bullet} &= \mathcal{A}^{\square \rightarrow \square} + \mathcal{A}^{\square \rightarrow \blacksquare} + \mathcal{A}^{\blacksquare \rightarrow \square} + \mathcal{A}^{\blacksquare \rightarrow \blacksquare},\end{aligned}$$

so by the dissymmetry theorem we have

$$\mathcal{A} + \mathcal{A}^{\square \rightarrow \square} + \mathcal{A}^{\square \rightarrow \blacksquare} + \mathcal{A}^{\blacksquare \rightarrow \square} + \mathcal{A}^{\blacksquare \rightarrow \blacksquare} \simeq \mathcal{A}^\square + \mathcal{A}^\blacksquare + \mathcal{A}^{\square \rightarrow \square} + \mathcal{A}^{\square \rightarrow \blacksquare} + \mathcal{A}^{\blacksquare \rightarrow \square}.$$

Since leaves have degree 1, we see that $\mathcal{A}^\square \simeq \mathcal{A}^{\square \rightarrow \blacksquare} \simeq \mathcal{A}^{\blacksquare \rightarrow \square} \simeq \mathcal{A}^{\blacksquare \rightarrow \blacksquare}$, so it follows that

$$\mathcal{A} + \mathcal{A}^{\square \rightarrow \square} + \mathcal{A}^{\blacksquare \rightarrow \blacksquare} \simeq \mathcal{A}^\blacksquare + \mathcal{A}^{\square \rightarrow \square} + \mathcal{A}^{\blacksquare \rightarrow \blacksquare}.$$

Finally, the classes $\mathcal{A}^{\square \rightarrow \square}$ and $\mathcal{A}^{\square \rightarrow \blacksquare}$ are either both empty or both contain a single graph of size 2 (depending on whether or not \mathcal{A} contains the tree with two vertices), so $\mathcal{A}^{\square \rightarrow \square} \simeq \mathcal{A}^{\square \rightarrow \blacksquare}$, and hence

$$\mathcal{A} + \mathcal{A}^{\blacksquare \rightarrow \blacksquare} \simeq \mathcal{A}^\blacksquare + \mathcal{A}^{\blacksquare \rightarrow \blacksquare}.$$

\square

3.2 Boltzmann sampler by center-rejection

In this section, we introduce a new technique for sampling from an unrooted class of trees that is specified by the dissymmetry theorem. As mentioned previously, it is possible to recursively build a Boltzmann sampler for any decomposable combinatorial class. Unfortunately, the equation

$$\mathcal{A} \simeq \mathcal{A}^\bullet + \mathcal{A}^{\bullet \rightarrow \bullet} - \mathcal{A}^{\bullet \rightarrow \bullet}$$

given by the dissymmetry theorem is not a true symbolic specification, because at first glance there is no combinatorial meaning or Boltzmann sampler rule that can be ascribed to the subtraction of the final term.

We overcome this difficulty by describing a Boltzmann sampler rule that accounts for this subtraction. More specifically, we show how to build a Boltzmann sampler for an arbitrary class that is specified by the dissymmetry theorem, assuming that there exist samplers for the corresponding vertex-rooted and edge-rooted classes. The sampler draws repeatedly from the class $\mathcal{A}^\bullet + \mathcal{A}^{\bullet \rightarrow \bullet}$, each

time obtaining a pair (T, r) where T is a tree and r is either a vertex or edge of T that is marked, and stops once this marked vertex/edge happens to be the center $c(T)$ of the tree (which can also be either a vertex or an edge). Thus it utilizes the technique of *rejection* – sampling from a superclass until the sampled object has a certain property [17]. As we will see, its correctness follows almost immediately from the main idea of the proof of the dissymmetry theorem.

Theorem 3.4. *Let \mathcal{A} be an unrooted class of trees, and suppose that we have Boltzmann samplers $\Gamma\mathcal{A}^\bullet(z)$ and $\Gamma\mathcal{A}^{\bullet-\bullet}(z)$. Then the following procedure is a Boltzmann sampler for \mathcal{A} :*

```

 $\Gamma\mathcal{A}(z)$ :
  do
    if  $\text{Bern}\left(\frac{\mathcal{A}^\bullet(z)}{\mathcal{A}^\bullet(z) + \mathcal{A}^{\bullet-\bullet}(z)}\right) = 1$  then
       $(T, r) \leftarrow \Gamma\mathcal{A}^\bullet(z)$ 
    else
       $(T, r) \leftarrow \Gamma\mathcal{A}^{\bullet-\bullet}(z)$ 
  until  $r = c(T)$ 
  return  $T$ 

```

Proof. Let T be an element of \mathcal{A} . To show that the above procedure is a Boltzmann sampler for \mathcal{A} , it suffices to show that the probability that it draws T is

$$\mathbb{P}_z[T] = \frac{z^{|T|}}{\mathcal{A}(z)}.$$

Let

$$\mathcal{A}_v(z) = \sum_{\substack{T' \in \mathcal{A} \\ c(T') \text{ is} \\ \text{a vertex}}} z^{|T'|} \quad \text{and} \quad \mathcal{A}_e(z) = \sum_{\substack{T' \in \mathcal{A} \\ c(T') \text{ is} \\ \text{an edge}}} z^{|T'|}.$$

Since each element of \mathcal{A} has either a vertex or an edge as its center, we have

$$\mathcal{A}(z) = \mathcal{A}_v(z) + \mathcal{A}_e(z).$$

We consider two cases: when $c(T)$ is a vertex, and when it is an edge. Let P denote the procedure inside the loop. In the first case,

$$\begin{aligned}
\mathbb{P}_z[T] &= \frac{\mathbb{P}[(T, c(T)) \text{ is drawn by } P]}{\mathbb{P}[T', c(T')] \text{ is drawn by } P \text{ for some tree } T'} \\
&= \frac{\frac{\mathcal{A}^\bullet(z)}{\mathcal{A}^\bullet(z) + \mathcal{A}^{\bullet-\bullet}(z)} \cdot \frac{z^{|T|}}{\mathcal{A}^\bullet(z)}}{\frac{\mathcal{A}^\bullet(z)}{\mathcal{A}^\bullet(z) + \mathcal{A}^{\bullet-\bullet}(z)} \cdot \frac{\mathcal{A}_v(z)}{\mathcal{A}^\bullet(z)} + \frac{\mathcal{A}^{\bullet-\bullet}(z)}{\mathcal{A}^\bullet(z) + \mathcal{A}^{\bullet-\bullet}(z)} \cdot \frac{\mathcal{A}_e(z)}{\mathcal{A}^{\bullet-\bullet}(z)}} \\
&= \frac{\frac{z^{|T|}}{\mathcal{A}^\bullet(z) + \mathcal{A}^{\bullet-\bullet}(z)}}{\frac{\mathcal{A}_v(z) + \mathcal{A}_e(z)}{\mathcal{A}^\bullet(z) + \mathcal{A}^{\bullet-\bullet}(z)}} \\
&= \frac{z^{|T|}}{\mathcal{A}(z)}
\end{aligned}$$

The second case follows by the same argument, except that the initial expression for the quantity $\mathbb{P}[(T, c(T)) \text{ is drawn by } P]$ is

$$\frac{\mathcal{A}^{\bullet-\bullet}(z)}{\mathcal{A}^\bullet(z) + \mathcal{A}^{\bullet-\bullet}(z)} \cdot \frac{z^{|T|}}{\mathcal{A}^{\bullet-\bullet}(z)}.$$

□

Finding the center of a tree. We recall from Definition 3.2 that the center $c(T)$ of a tree T is the vertex or edge of T that is obtained by the iterative process that, at each step, deletes all leaves of T , and that halts when T has size less than 3. This can be computed by the following linear-time algorithm. Initialize a FIFO queue Q that contains the leaves of T , and then repeatedly: pop a leaf ℓ from Q , delete ℓ from T , and push the former neighbor of ℓ into Q if it is now a leaf (*i.e.* if its new degree is 1). Continue until Q has either one vertex, or two vertices that are connected by an edge. Since any newly-created leaf will not be popped until all already-existent leaves are handled, this algorithm mimics the process of deleting all leaves simultaneously at each step, and since each vertex is popped from Q at most one time, the algorithm runs in linear time in the number of vertices of T .

Lemma 3.5. *The number of iterations made by the do-while loop until a suitable tree is drawn (*i.e.* one that is marked at its center) is, on average,*

$$E_{\mathcal{A}}(z) = \frac{\mathcal{A}^{\bullet}(z) + \mathcal{A}^{\bullet\rightarrow\bullet}(z)}{\mathcal{A}(z)}.$$

Proof. The probability of exiting the loop on a given round is the chance of drawing a tree rooted at its center, which is

$$\mathbb{P}[T', c(T')] \text{ is drawn by } P \text{ for some tree } T'].$$

Furthermore, we saw in the proof of Theorem 3.4 that this probability is

$$p = \frac{\mathcal{A}^{\bullet}(z)}{\mathcal{A}^{\bullet}(z) + \mathcal{A}^{\bullet\rightarrow\bullet}(z)} \cdot \frac{\mathcal{A}_v(z)}{\mathcal{A}^{\bullet}(z)} + \frac{\mathcal{A}^{\bullet\rightarrow\bullet}(z)}{\mathcal{A}^{\bullet}(z) + \mathcal{A}^{\bullet\rightarrow\bullet}(z)} \cdot \frac{\mathcal{A}_e(z)}{\mathcal{A}^{\bullet\rightarrow\bullet}(z)} = \frac{\mathcal{A}(z)}{\mathcal{A}^{\bullet}(z) + \mathcal{A}^{\bullet\rightarrow\bullet}(z)}.$$

Since the number of rounds of the loop is a geometric random variable with success probability p , its expected value is

$$\frac{1}{p} = \frac{\mathcal{A}^{\bullet}(z) + \mathcal{A}^{\bullet\rightarrow\bullet}(z)}{\mathcal{A}(z)}.$$

□

3.3 Example

Let \mathcal{T} be the class of unlabeled, unrooted, non-plane 2-3 trees – equivalently, the class of trees whose vertices each have degree 1, 3, or 4 (notice that if you hang such a tree from one of its leaves, each internal node has either 2 or 3 children). In order to analyze \mathcal{T} with the dissymmetry theorem, we begin by determining combinatorial specifications for the classes \mathcal{T}^{\bullet} , $\mathcal{T}^{\bullet\rightarrow\bullet}$, and $\mathcal{T}^{\bullet\rightarrow\bullet}$; respectively, the class of trees in \mathcal{T} where one vertex is marked, the class of trees in \mathcal{T} where one undirected edge is marked, and the class of trees in \mathcal{T} where one directed edge is marked.

Objects in \mathcal{T}^{\bullet} are decomposed at their root – indeed, an object in \mathcal{T}^{\bullet} is a (marked) root node with 1, 3, or 4 neighbors, where each neighbor is a node with 0, 2, or 3 other neighbors, so we have the decomposition

$$\begin{aligned} \mathcal{T}^{\bullet} &= \mathcal{Z}^{\bullet} \times \text{SET}_{1,3,4}(\mathcal{S}), \\ \mathcal{S} &= \mathcal{Z} + \mathcal{Z} \times \text{SET}_{2,3}(\mathcal{S}). \end{aligned}$$

Objects in $\mathcal{T}^{\bullet\rightarrow\bullet}$ and $\mathcal{T}^{\bullet\rightarrow\bullet}$ are decomposed at their marked edge – in the first case, the edge has a set of two neighbors, while in the second case it has a sequence of two neighbors (since the head and tail of the edge are distinguished), so we have

$$\begin{aligned} \mathcal{T}^{\bullet\rightarrow\bullet} &= \text{SET}_2(\mathcal{S}), \\ \mathcal{T}^{\bullet\rightarrow\bullet} &= \mathcal{S} \times \mathcal{S}. \end{aligned}$$

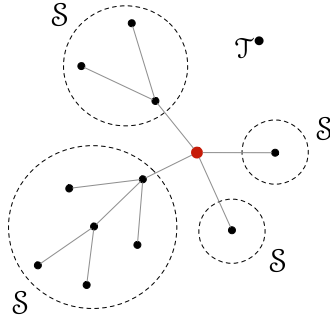


Figure 3. Decomposition of vertex-rooted 2-3 trees.

Using the Maple package `combstruct`, we compute the ordinary generating functions for these classes:

$$\mathcal{T}^\bullet(z) = z^2 + 2z^4 + 2z^5 + 2z^6 + 4z^7 + 6z^8 + 10z^9 + 17z^{10} + 29z^{11} + 48z^{12} + 85z^{13} + 148z^{14} + 259z^{15} + \dots$$

$$\mathcal{T}^{\bullet-\bullet}(z) = z^2 + z^4 + z^5 + 2z^6 + 3z^7 + 5z^8 + 8z^9 + 14z^{10} + 24z^{11} + 42z^{12} + 73z^{13} + 131z^{14} + 230z^{15} + \dots$$

$$\mathcal{T}^{\bullet\rightarrow\bullet}(z) = z^2 + 2z^4 + 2z^5 + 3z^6 + 6z^7 + 9z^8 + 16z^9 + 27z^{10} + 48z^{11} + 82z^{12} + 146z^{13} + 259z^{14} + 460z^{15} + \dots$$

and by Theorem 3.1 we have

$$\begin{aligned} \mathcal{T}(z) &= \mathcal{T}^\bullet(z) + \mathcal{T}^{\bullet-\bullet}(z) - \mathcal{T}^{\bullet\rightarrow\bullet}(z) \\ &= z^2 + z^4 + z^5 + z^6 + z^7 + 2z^8 + 2z^9 + 4z^{10} + 5z^{11} + 8z^{12} + 12z^{13} + 20z^{14} + 29z^{15} + \dots \end{aligned}$$

In order to build a Boltzmann sampler for \mathcal{T} using Theorem 3.4, we require samplers for \mathcal{T}^\bullet and $\mathcal{T}^{\bullet-\bullet}$. It is known how to construct these samplers [29], and we will provide more detail on the necessary tools in Section 4.4.1. However, we can immediately use Lemma 3.5 to estimate the rejection cost for different values of z , as shown in Table 2.

z	Expected size	Probability of success on each iteration	Expected number of iterations of the loop
0.01	2.000	0.5000	2.000
0.1	2.023	0.4972	2.011
0.5	3.647	0.3043	3.287
$\rho \approx 0.508256$	4.224	0.2455	4.051

Table 2. Rejection cost for the dissymmetry theorem Boltzmann sampler for \mathcal{T} .

In Figure 4 we show a plot of the expected number of iterations as a function of $z\mathcal{A}'(z)/\mathcal{A}(z)$, the expected size of the returned object. It appears that these quantities have an approximately linear relationship with with a slope of about 1.

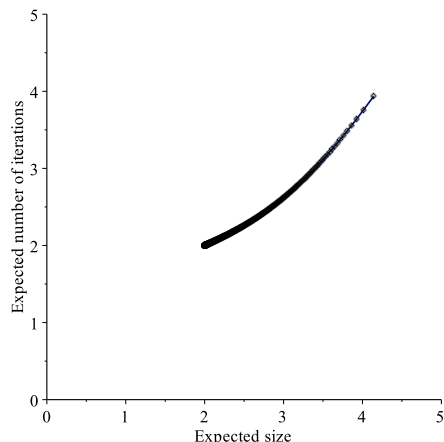


Figure 4. Rejection cost vs. expected size for the dissymmetry theorem Boltzmann sampler for \mathcal{T} .

4 Cycle pointing

In Section 3, we saw that the dissymmetry theorem allows one to enumerate and create a Boltzmann sampler for an unlabeled class of trees by analyzing three corresponding rooted classes. However, this technique lacks combinatorial strength – it relies on the very particular notion of the center of a tree to establish a bijection, but fails to provide a symbolic decomposition that carries along with it all the usual trappings such as automatic Boltzmann samplers and asymptotics. Furthermore, our sampler that emerges from the dissymmetry theorem might not be ideal, because it relies on a potentially costly rejection process. Finally, the existence of a method for pointing at labeled objects that establishes a 1-to- n correspondence between the unpointed and pointed classes (*cf.* Section 2.3.1) raises the question of whether or not such a method exists for unlabeled objects as well.

Cycle pointing, a technique introduced by Bodirsky *et al.* [18, 19], addresses these issues simultaneously. Instead of selecting a certain distinguished vertex of a graph, one selects a *cycle* of vertices satisfying a certain property, and this property is chosen in such a way that there are exactly n pointed objects of size n for each object of size n in the original class. Together with the ability to decompose the pointed class, this immediately provides an unbiased sampler for the unlabeled, unpointed class that does not use rejection. As we will mention again later, cycle pointing in the labeled case exactly reduces to vertex pointing, so cycle pointing can be thought of as a generalization of the method of Section 2.3.1 to the unlabeled case.

However, using cycle pointing to enumerate and build a sampler for a class is quite a challenging task; two of the main contributions of this work are to help break down this process for future readers, and to apply cycle pointing to enumerate and build the first unbiased samplers for the classes of distance-hereditary and three-leaf power graphs.

Outline. We first give a refresher on certain relevant graph theoretic notions, and we then dive into the definitions and main results of cycle pointing in Section 4.1. Section 4.2 shows how to write a combinatorial specification for a cycle-pointed class, Section 4.3 shows how to exploit the decomposition using elements of Pólya theory to derive an enumeration for the class, and Section 4.4 shows how to automatically translate the decomposition into an unbiased sampler for the underlying class.

4.1 Introduction

We begin this section by reviewing some definitions from graph theory, which once in hand allow us to introduce cycle pointing and present the relevant results from Bodirsky *et al.* [18, 19]. All proofs in Section 4 are adapted from these two sources.

Definition 4.1. An *automorphism* of a graph G is a mapping from G to itself that preserves its underlying structure, in particular its adjacencies and nonadjacencies. More formally, an automorphism of G is a bijection $\phi : V(G) \rightarrow V(G)$ such that for any $u, v \in V(G)$, $uv \in E(G)$ iff $\phi(u)\phi(v) \in E(G)$.

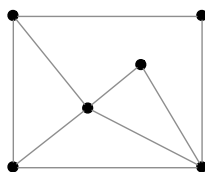


Figure 5. An identity graph.

Every graph has at least one automorphism – the identity map – and graphs with no other automorphisms are called *identity graphs*. All other graphs are said to have *non-trivial* automorphisms. With the operation of composition, the set of automorphisms of a graph forms a group called its *automorphism group*.

Since each automorphism is a permutation of the vertices of G , it may be uniquely decomposed as a set of disjoint cycles on the vertices of G . For example, the automorphism shown in Figure 6 has three cycles, shown in different colors. Figure 7 shows a different automorphism on the same graph.

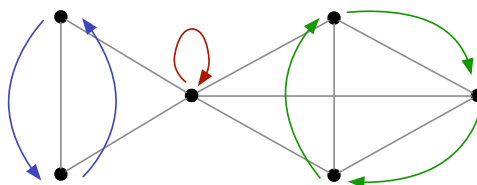


Figure 6. An automorphism of a graph, with cycles shown.

From this concept of an automorphism and its cycles, we can define cycle-pointed structures and the cycle pointing operator:

Definition 4.2. For a graph G , a *cycle-pointed structure* is a pair $P = (G, c)$ such that there exists at least one automorphism of G having c as one of its cycles (such an automorphism is called a *c-automorphism* of G). Then c is called the *marked cycle* of P , and G is called its *underlying structure*. P is called *symmetric* if c has at least two vertices (because in this case c corresponds to a non-trivial automorphism of G).

For a better intuition of what constitutes a cycle-pointed structure, we note that Figure 8 is a valid cycle-pointed structure, while Figure 9 is not. One way to see that Figure 9 is not a cycle-pointed structure is to note that, since automorphisms preserve adjacencies and nonadjacencies, any

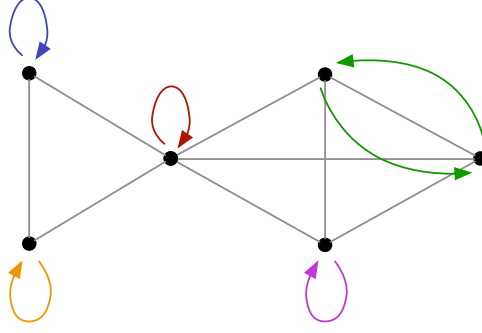


Figure 7. A different automorphism on the graph from Figure 6.

automorphism must map a vertex of a certain degree to a vertex of the same degree (this condition is necessary but of course not sufficient), so the red marked cycle cannot be part of any automorphism.

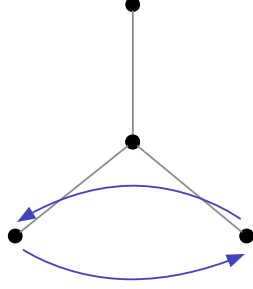


Figure 8. Valid cycle-pointed structure.

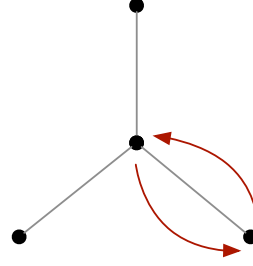


Figure 9. Invalid cycle-pointed structure.

Two cycle-pointed structures $P = (G, c)$ and $P' = (G', c')$ are considered isomorphic if there exists an isomorphism from G to G' that maps c to c' in a manner that preserves the cyclic order of the cycles. For example, in Figure 10, the first two cycle-pointed structures are isomorphic, while the third is not isomorphic to either. Beginning in Definition 4.3, we will consider, unless stated otherwise, two isomorphic cycle-pointed structures to be exactly the same.

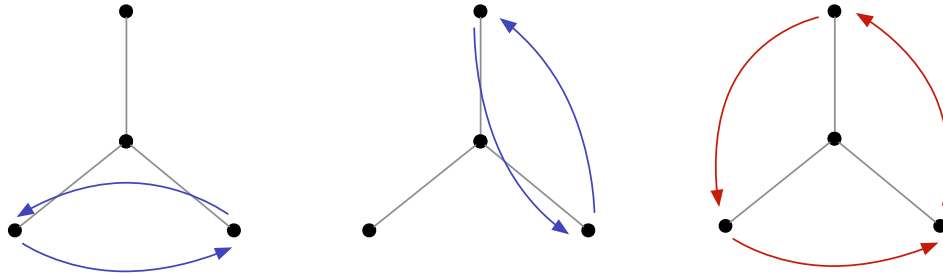


Figure 10. Two isomorphic cycle-pointed structures, and one counterexample.

Definition 4.3. For a class \mathcal{A} of graphs, the *cycle-pointed* class corresponding to \mathcal{A} is the class denoted \mathcal{A}° containing all (non-isomorphic) cycle-pointed structures whose underlying structure is in \mathcal{A} , where the size of an element (G, c) in \mathcal{A}° is defined as the size of G in \mathcal{A} . We refer to the map $\mathcal{A} \rightarrow \mathcal{A}^\circ$ as the *cycle-pointing operator*. Also more generally, we refer to any class of cycle-pointed structures as a cycle-pointed class.

Remark 4.4. The cycle-pointing operator, owing to its differential nature (cf. Theorem 4.5), satisfies the following compatibility properties:

- $(\mathcal{A} + \mathcal{B})^\circ = \mathcal{A}^\circ + \mathcal{B}^\circ$
- $(\mathcal{A} \times \mathcal{B})^\circ = \mathcal{A}^\circ \times \mathcal{B} + \mathcal{A} \times \mathcal{B}^\circ$

For a labeled graph, the only automorphism is the identity (due to the distinct labels on each vertex), so the only valid cycle-pointed structures have singleton cycles as their marked cycles. Thus in the labeled case, cycle pointing reduces to vertex pointing, so there are n cycle-pointed objects of size n for each unpointed object of size n .

As alluded to earlier, this 1-to- n correspondence carries over to the unlabeled case as well – for each unlabeled graph of size n , there are exactly n cycle-pointed structures of size n whose underlying structure is that graph. For example, Figure 11 shows the four cycle-pointed structures of the claw graph, and Figure 12 shows the four cycle-pointed structures of the square graph.

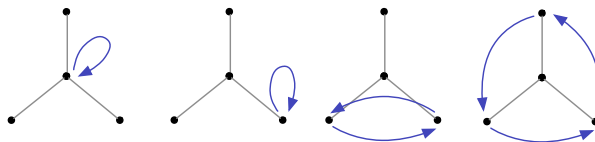


Figure 11. Cycle-pointed structures of the claw graph.

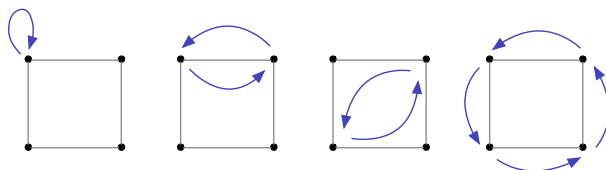


Figure 12. Cycle-pointed structures of the square graph.

Theorem 4.5. Let \mathcal{A} be an unlabeled class of graphs. Then for each graph $G \in \mathcal{A}$ of size n , there are exactly n objects of size n in \mathcal{A}° whose underlying structure is G . Thus, the OGF for \mathcal{A}° satisfies

$$\mathcal{A}^\circ(z) = z\mathcal{A}'(z).$$

In order to prove this theorem, we must first introduce a few new concepts.

Definition 4.6. For an unlabeled graph G , a *symmetry* of G is a pair (G^ℓ, σ) where G^ℓ is a labeled graph whose unlabeled structure is G and σ is an automorphism of G . The set of symmetries of G is denoted $\text{Sym}(G)$.

For example, Figure 13 shows the symmetries of the path graph on three vertices.

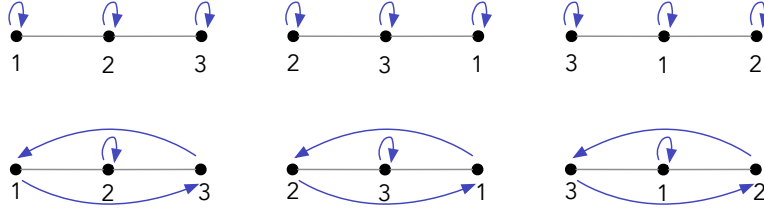


Figure 13. Symmetries of the path on three vertices.

Lemma 4.7. *An unlabeled graph G of size n has $n!$ symmetries.*

Proof. Let \mathcal{G} be the automorphism group of G , and let X be the set of $n!$ labeled graphs whose unlabeled structure is G , considered **not** up to isomorphism (so for example, the path 1-2-3 and the path 3-2-1 would be considered distinct in X , even though they are not distinct labeled graphs). Then we may define a group action of \mathcal{G} on X – essentially, a rule by which every element of \mathcal{G} maps each element of X to another element of X – by, for $\sigma \in \mathcal{G}$ and $x \in X$, defining $\sigma \cdot x$ to be the element of X obtained by applying the automorphism σ to the labeled graph x .

We say that two elements x, y are in the same orbit of this group action if there exists some $\sigma \in \mathcal{G}$ such that $\sigma \cdot x = y$; in other words, if there is an automorphism mapping x to y . Since \mathcal{G} is the automorphism group of G , the orbits X/\mathcal{G} of this action are in bijection with the labeled graphs whose unlabeled structure is G , considered up to isomorphism.

Since there is exactly one symmetry of G for each choice of one orbit of the action and one automorphism of G , the number of symmetries of G is

$$|\text{Sym}(G)| = |\mathcal{G}| \cdot |X/\mathcal{G}|.$$

By Burnside's lemma, this is equal to

$$\sum_{g \in \mathcal{G}} |\text{Fix}(g)|,$$

where $\text{Fix}(g)$ is the set of elements $x \in X$ such that $g \cdot x = x$ [20]. However the only element of \mathcal{G} that fixes any elements of X is the identity automorphism, and this fixes all $n!$ elements of X , so it follows that

$$|\text{Sym}(G)| = |\mathcal{G}| \cdot |X/\mathcal{G}| = n!.$$

□

Similarly, we define a symmetry of a cycle-pointed structure $P = (G, c)$ as a tuple (G^ℓ, c^ℓ, σ) such that (G^ℓ, σ) is a symmetry of G , c^ℓ is a labeled cycle whose unlabeled structure is c , and σ preserves the cycle c^ℓ . By a parallel argument to the one given above, every unlabeled cycle-pointed object of size n has $n!$ symmetries.

Definition 4.8. For an unlabeled cycle-pointed structure $P = (G, c)$, a *c-symmetry* of P is a tuple (G^ℓ, c^ℓ, σ) where G^ℓ is a labeled graph whose unlabeled structure is G , c^ℓ is a labeled cycle whose unlabeled structure is c , and σ is a c -automorphism of P (note that since the set of automorphisms having c as a cycle is a subset of the set of automorphisms that respect c , every c -symmetry of P is also a symmetry of P , but the reverse is not necessarily true). Furthermore, a *rooted c-symmetry* of P is a tuple $(G^\ell, c^\ell, \sigma, v)$ where (G^ℓ, c^ℓ, σ) is a c -symmetry of P and v is a vertex on c^ℓ . The set of rooted c -symmetries of P is denoted $\text{RSym}(P)$.

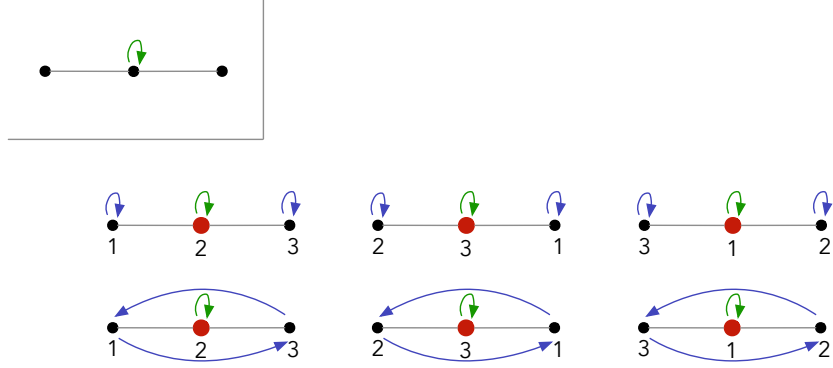


Figure 14. A cycle-pointed structure and its rooted c -symmetries.

For example, Figure 14 shows the rooted c -symmetries of a cycle-pointed structure (G, c) , where the cycle c is shown in green, and for each rooted c -symmetry $(G^\ell, c^\ell, \sigma, v)$, the cycles of σ that are not c^ℓ are shown in blue and the root v is shown in red.

Lemma 4.9. *An unlabeled cycle-pointed structure $P = (G, c)$ of size n has $n!$ rooted c -symmetries.*

Proof. Since P has $n!$ symmetries, it suffices to establish a bijection between the symmetries of P and the rooted c -symmetries of P . We will accomplish this by choosing some fixed c -symmetry of P , and then for each symmetry of P , using its “distance” from this fixed c -symmetry to map it to a rooted c -symmetry of P .

Fix a c -symmetry $(G_0^\ell, c_0^\ell, \sigma_0)$ of P (so c_0^ℓ is a cycle of σ_0), and consider an arbitrary symmetry (G^ℓ, c^ℓ, σ) of P (so σ preserves c^ℓ , but c^ℓ is not necessarily one of its cycles). Since σ preserves the cycle c^ℓ , it shifts the atoms $(v_1, \dots, v_{|c^\ell|})$ of c^ℓ by some value $1 \leq r \leq |c^\ell|$ modulo $|c^\ell|$, mapping v_i to $v_{i+r \pmod{|c^\ell|}}$. Also, since (G_0^ℓ, c_0^ℓ) and (G^ℓ, c^ℓ) are labeled cycle-pointed structures with the same unlabeled structure, there is an isomorphism σ_1 from G^ℓ to G_0^ℓ that maps c^ℓ to c_0^ℓ .

Let $\tau = \sigma_1^{-1} \sigma_0 \sigma_1$, and let v be the atom on c^ℓ having the r^{th} smallest label. Then we claim that $(G^\ell, c^\ell, \tau^{-r+1} \sigma, v)$ is a rooted c -symmetry of P . To show this, it suffices to show that $\tau^{-r+1} \sigma$ has c^ℓ as one of its cycles. Indeed, since σ_1 maps c^ℓ to c_0^ℓ , σ_0 has c_0^ℓ as one of its cycles, and σ_1^{-1} maps c_0^ℓ to c^ℓ , it follows that τ has c^ℓ as one of its cycles. Thus τ^{-1} shifts c^ℓ backwards by one step, and since σ shifts c^ℓ forward by r steps it follows that $(\tau^{-1})^{r-1} \sigma = \tau^{-r+1} \sigma$ shifts c^ℓ forward by one step and hence has c^ℓ as one of its cycles.

In the reverse direction, for a given rooted c -symmetry $(G^\ell, c^\ell, \sigma', v)$, let $r \geq 1$ be such that v has the r^{th} smallest label on c^ℓ . Then defining σ_1 as the first paragraph, we recover $\sigma = (\sigma_1^{-1} \sigma_0 \sigma_1)^{r-1} \sigma'$, and by the reverse argument to the one above it follows that (G^ℓ, c^ℓ, σ) is a symmetry of P . \square

Lemma 4.10. *For a given symmetry (G^ℓ, σ) of an unlabeled graph G of size n , there are n rooted c -symmetries that have σ as their automorphism and G^ℓ as their graph.*

Proof. Since G^ℓ has n vertices and each vertex is on only one cycle of σ , there can be at most n suitable rooted c -symmetries. On the other hand, there is indeed one for each vertex v of G^ℓ , obtained by choosing c^ℓ as the cycle of σ containing v . \square

With these results in hand, we return to the proof of Theorem 4.5.

Theorem. *Let \mathcal{A} be an unlabeled class of graphs. Then for each graph $G \in \mathcal{A}$ of size n , there are exactly n objects of size n in \mathcal{A}° whose underlying structure is G . Thus, the OGF for \mathcal{A}° satisfies*

$$\mathcal{A}^\circ(z) = z\mathcal{A}'(z).$$

Proof. Let $G \in \mathcal{A}$ be a graph of size n , and let S be the set of objects in \mathcal{A}° whose underlying structure is G . From Lemma 4.7 we know that $|\text{Sym}(G)| = n!$, and from Lemma 4.9 we know that each element of S has $n!$ rooted c -symmetries. From Lemma 4.10, each symmetry of G gives rise to exactly n rooted c -symmetries, and since the cycle-pointed structure of each of these rooted c -symmetries is in S it follows that $n \cdot |\text{Sym}(G)| = |S| \cdot n!$. Since $|\text{Sym}(G)| = n!$, $|S| = n$. \square

Corollary 4.11. *For a class \mathcal{A} of unlabeled graphs, $\mathcal{A}_n = \frac{1}{n} \mathcal{A}_n^\circ$. Furthermore, if $\Gamma \mathcal{A}^\circ(z)$ is a Boltzmann sampler for \mathcal{A}° , then*

$$\tilde{\Gamma} \mathcal{A}(z) = ((G, c) \leftarrow \Gamma \mathcal{A}^\circ(z); \text{ \texttt{return} } G);$$

is an unbiased sampler for \mathcal{A} , in the sense that

$$\mathbb{P}_z[G \mid |G| = n] = \frac{1}{\mathcal{A}_n}.$$

Proof. The first claim is a restatement of Theorem 4.5, and the second claim follows by a similar argument to the one given in Lemma 2.4. \square

4.2 Decomposition of cycle-pointed classes

In order to analyze an unlabeled class \mathcal{A} with Corollary 4.11, it is first necessary to enumerate and build a Boltzmann sampler for the cycle-pointed class \mathcal{A}° , and the first step in accomplishing this is to determine a symbolic specification for this class. While objects in \mathcal{A} lack a distinguished feature at which they can be decomposed, objects in \mathcal{A}° have such a feature – the marked cycle – and it is this cycle which is used to decompose the graphs and develop a specification.

The details of the decomposition technique vary depending on the nature of the class \mathcal{A} ; as we are primarily interested in classes of trees, we will defer consideration of the decomposition techniques used for non-tree cycle-pointed graph classes, and instead focus on the techniques used for cycle-pointed classes of trees.

4.2.1 Theory

Let \mathcal{A} be an unlabeled class of trees, and let \mathcal{A}° be its cycle-pointed class. To decompose \mathcal{A}° , we begin by partitioning it into two classes

$$\mathcal{A}^\circ = \mathcal{A}^\bullet + \mathcal{A}^\circledast$$

where the first contains all elements of \mathcal{A}° whose marked cycle has length 1, and the second, called the *symmetric cycle-pointed* class of \mathcal{A} , contains all elements whose marked cycle has length at least 2 (*i.e.* all symmetric elements of \mathcal{A}°). By considering a marked singleton cycle on a vertex as equivalent to marking the vertex itself, we may consider the class \mathcal{A}^\bullet to be the vertex-rooted class of \mathcal{A} .

Elements in the first class, \mathcal{A}^\bullet , are decomposed at their pointed vertex (similarly to how Cayley trees were decomposed in Section 2), which is generally a straightforward process. Decomposing elements in the second class is more challenging, and for this we introduce two new concepts: the *cycle-pointed substitution operator*, and the *center of symmetry* of a cycle-pointed tree. This center of symmetry provides the distinguished location at which a symmetric cycle-pointed tree can be decomposed, and the cycle-pointed substitution operator allows us to decompose a cycle-pointed tree at its center of symmetry.

A note on notation. The concepts of cycle-pointed and symmetric cycle-pointed classes were introduced by Bodirsky *et al.* [19], who use the notations \mathcal{A}° and \mathcal{A}^\circledast (in [18], \mathcal{A}^\bullet is used in place of \mathcal{A}°). We reintroduce them here with slightly different notations – \mathcal{A}° and \mathcal{A}^\circledast – for the sake of clarity. We have chosen these both to provide a visual representation that we are marking *cycles*, and to minimize the chance of confusion with other common uses of \mathcal{A}° and \mathcal{A}^\bullet (e.g. vertex pointing).

Definition 4.12. Let c be a cycle of length k , and let $c_1 = (v_{11}, \dots, v_{1k}), \dots, c_\ell = (v_{\ell 1}, \dots, v_{\ell k})$ be ℓ isomorphic copies of c (on disjoint sets of vertices). Then the composition $c_1 \circ \dots \circ c_\ell$ of these cycles is defined as the cycle

$$(v_{11}, v_{21}, \dots, v_{\ell 1}, v_{12}, v_{22}, \dots, v_{\ell 2}, \dots, v_{1k}, v_{2k}, \dots, v_{\ell k}).$$

Definition 4.13. Let \mathcal{B}° (or \mathcal{B}^\circledast , etc.) be a cycle-pointed class, and let \mathcal{C} be a *non* cycle-pointed class. Then the *cycle-pointed substitution* $\mathcal{B}^\circ \odot \mathcal{C}$ is defined as the cycle-pointed class containing all structures obtained as follows:

1. Let $P = (G, c)$ be an element of \mathcal{B}° , and let $c = (v_1, \dots, v_k)$.
2. Replace the vertices of G that are on c with elements of \mathcal{C}° , and those that are not on c with elements of \mathcal{C} , in a manner that respects at least one c -automorphism of P (we say that a replacement *respects* an automorphism σ if for any vertex v of G , the structures that replace v and $\sigma(v)$ are isomorphic). Note that the vertices v_1, \dots, v_k of c must be replaced with isomorphic copies $Q_1 = (H_1, c'_1), \dots, Q_k = (H_k, c'_k)$ of the same cycle-pointed structure $Q = (H, c')$.
3. Let the marked cycle of the composed structure be $c'_1 \circ \dots \circ c'_k$.

For example, letting \mathcal{B} be the class of trees, Figure 15 illustrates the construction of an element of $\mathcal{B}^\circ \odot \mathcal{B}$, where the element $P = (T, c)$ from \mathcal{B}° is depicted with hollow vertices and a blue cycle, the cycles c'_1 and c'_2 are shown in red, and the composed cycle is shown in green. Note that in order for the substitution to respect a (in fact, the only) c -automorphism of P , the graphs substituted at the two bottom vertices of G must be isomorphic, and the graphs substituted at the two middle vertices of G must be as well.

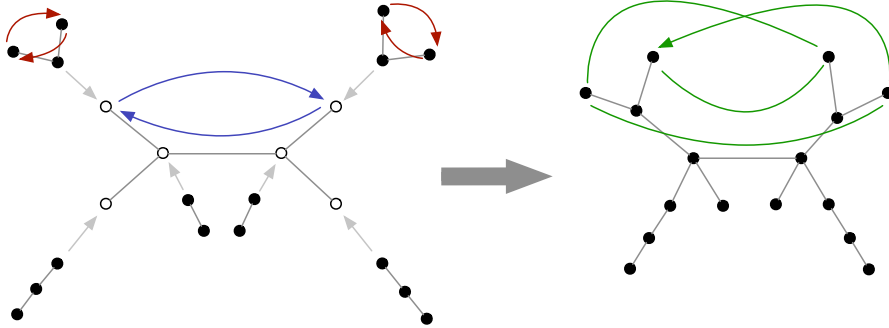


Figure 15. A cycle-pointed substitution.

Lemma 4.14. Let $P = (T, c)$ be a symmetric cycle-pointed tree. For each consecutive pair of vertices (v_i, v_{i+1}) on c , call the path p_i through T that connects these vertices a connecting path of P (note: all indices are taken modulo $|c|$). Then all connecting paths of P share the same middle element, called the center of symmetry of P , which is either a vertex (if the paths all have even length) or an edge (if they all have odd length).

Proof. Let U be the subgraph of T formed by the union of all the paths p_i , and let σ be a c -automorphism of T (such a σ must exist by Definition 4.2). Since p_i is connected to p_{i+1} at v_{i+1} for each i , it follows that p_i is connected to p_j for any two indices i and j , so U is a connected subgraph of T . Hence, U is a tree. Furthermore, since c is a cycle of σ , σ maps each connecting path to another connecting path (indeed it maps p_i to p_{i+1}), so U is fixed by σ and hence $\sigma|_U$ (σ restricted to U) is an automorphism of U .

Let $x = c(U)$ be the center of U (cf. Definition 3.2). We claim that x is the middle element of all connecting paths of P . First, since the iterative procedure which defines the center of a tree is invariant to automorphism, it follows that x is fixed by any automorphism of U . Since $\sigma|_U$ is a c -automorphism of U , the cyclic group $\mathcal{G} = \langle \sigma|_U \rangle$ acts transitively on the set of vertices of c , and since x is fixed by all elements of \mathcal{G} it follows that x is equidistant from all vertices of c . Furthermore, since $\sigma|_U$ is a c -automorphism of U , \mathcal{G} acts transitively on the set of paths p_i , and since x is fixed by all elements of \mathcal{G} and is on at least one p_i (after all, it is in U), it follows that x is on all the p_i . Combining these two results, it follows that x is the middle of all connecting paths of P . \square

Informally, the center of symmetry of a cycle-pointed tree is obtained by first deleting all vertices and edges that are not on some path connecting two vertices of the marked cycle, and then taking the traditional center of the resulting tree. Thus it can be thought of as the center of the marked cycle, and it is fixed by any automorphism of the tree that respects the marked cycle.

With these tools in hand, we use the following procedure to decompose a cycle-pointed class \mathcal{A}° of trees:

1. Write \mathcal{A}° as $\mathcal{A}^\circ = \mathcal{A}^\bullet + \mathcal{A}^\mathcal{C}$.
2. Decompose an object in \mathcal{A}^\bullet at its pointed vertex.
3. Write $\mathcal{A}^\mathcal{C} = \mathcal{A}_v^\mathcal{C} + \mathcal{A}_e^\mathcal{C}$, where $\mathcal{A}_v^\mathcal{C}$ ($\mathcal{A}_e^\mathcal{C}$, respectively) contains the elements of $\mathcal{A}^\mathcal{C}$ whose center of symmetry is a vertex (edge, respectively).
4. Decompose a structure in $\mathcal{A}_v^\mathcal{C}$ at its center of symmetry, v . Since v is the center of symmetry and the marked cycle has length at least 2, v is attached to at least two isomorphic copies of the same tree across which the marked cycle can be decomposed into isomorphic cycles, as well as possibly other trees with no marked cycle. Thus the neighbors of v can be accounted for by substituting into $\text{SET}^\mathcal{C}/\text{CYC}^\mathcal{C}$ if \mathcal{A} is a non-plane/plane class (with restrictions on the size of the set/cycle if \mathcal{A} has restrictions on the degrees of its vertices).
5. Decompose a structure in $\mathcal{A}_e^\mathcal{C}$ at its center of symmetry, e . Since e is the center of symmetry and the marked cycle of the structure has length at least 2, the trees attached to the two endpoints of e must be two isomorphic copies of the same tree, and the marked cycle must be a composition of two isomorphic cycles on these trees. Thus, these can be accounted for by substituting into $\text{SET}_2^\mathcal{C}$.

4.2.2 Example

Recall the class \mathcal{T} of unlabeled, unrooted, non-plane 2-3 trees, which we analyzed using the dissymmetry theorem in Section 3.3. In order to decompose \mathcal{T}° , we begin by partitioning the class into its symmetric and non-symmetric parts:

$$\mathcal{T}^\circ = \mathcal{T}^\bullet + \mathcal{T}^\mathcal{C}.$$

Objects in \mathcal{T}^\bullet are decomposed at their root (the vertex of the marked singleton cycle) – indeed, an object in \mathcal{T}^\bullet is a root node with 1, 3, or 4 neighbors, where each neighbor is a node with 0, 2, or 3 other neighbors, so we have the decomposition

$$\mathcal{T}^\bullet = \mathbb{Z}^\bullet \times \text{SET}_{1,3,4}(\mathcal{S})$$

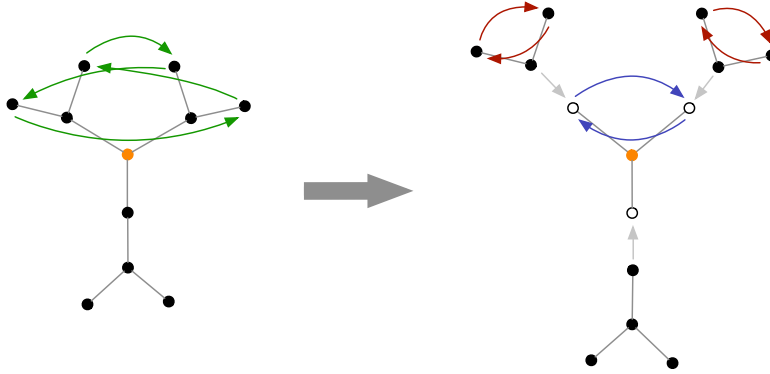


Figure 16. A cycle-pointed tree decomposed at its center of symmetry, which is the orange vertex.

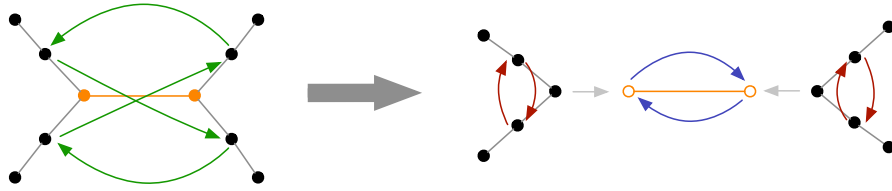


Figure 17. A cycle-pointed tree decomposed at its center of symmetry, which is the orange edge.

$$\mathcal{S} = \mathcal{Z} + \mathcal{Z} \times \text{SET}_{2,3}(\mathcal{S}).$$

Next, $\mathcal{T}^{\mathcal{S}}$ is partitioned as

$$\mathcal{T}^{\mathcal{S}} = \mathcal{T}_v^{\mathcal{S}} + \mathcal{T}_e^{\mathcal{S}}.$$

Objects in $\mathcal{T}_v^{\mathcal{S}}$ are decomposed at their center of symmetry, which is a vertex – indeed, an object in $\mathcal{T}_v^{\mathcal{S}}$ is a node (the center of symmetry) with a symmetric cycle-pointed set of either 3 or 4 elements of \mathcal{S} (the center of symmetry cannot be a leaf since the marked cycle has length at least 2), so we have the decomposition

$$\mathcal{T}_v^{\mathcal{S}} = \mathcal{Z} \times (\text{SET}_{3,4}^{\mathcal{S}} \odot \mathcal{S}).$$

Finally, objects in $\mathcal{T}_e^{\mathcal{S}}$ are decomposed at their center of symmetry, which is an edge – indeed, an object in $\mathcal{T}_e^{\mathcal{S}}$ is an edge (the center of symmetry) whose extremities form a symmetric cycle-pointed set of 2 elements of \mathcal{S} (it must be symmetric since the marked cycle of objects in $\mathcal{T}_e^{\mathcal{S}}$ has length at least 2), so we have the decomposition

$$\mathcal{T}_e^{\mathcal{S}} = \text{SET}_2^{\mathcal{S}} \odot \mathcal{S}.$$

Combining these results, we have the system of equations

$$\mathcal{T}^{\mathcal{C}} = \mathcal{Z}^{\bullet} \times \text{SET}_{1,3,4}(\mathcal{S}) + \mathcal{Z} \times (\text{SET}_{3,4}^{\mathcal{S}} \odot \mathcal{S}) + \text{SET}_2^{\mathcal{S}} \odot \mathcal{S}$$

$$\mathcal{S} = \mathcal{Z} + \mathcal{Z} \times \text{SET}_{2,3}(\mathcal{S}).$$

4.3 Enumeration

4.3.1 Theory

After obtaining a combinatorial specification for \mathcal{A}° , the next step is to use this specification to derive a system of generating function equations for the OGF $\mathcal{A}^\circ(z)$, and from there to compute the coefficients \mathcal{A}_n° and $\mathcal{A}_n = \frac{1}{n} \mathcal{A}_n^\circ$. In order to translate such a specification to a generating function equation, we use *cycle index sums*, first introduced by Pólya [11]. These are generalizations of ordinary generating functions that capture all symmetries of the objects in a combinatorial class within a single power series over infinitely many variables.

Definition 4.15. If \mathcal{B} is a non cycle-pointed class, the *cycle index sum* of \mathcal{B} is defined as the following formal power series over the variables s_1, s_2, \dots :

$$Z_{\mathcal{B}}(s_1, s_2, \dots) = \sum_{G \in \mathcal{B}} \sum_{(G^\ell, \sigma) \in \text{Sym}(G)} \frac{1}{|G|!} \prod_{i=1}^{|G|} s_i^{c_i(\sigma)}$$

where $c_i(\sigma)$ is defined as the number of cycles in σ of length i . If \mathcal{B} is a cycle-pointed class, the *cycle index sum* of \mathcal{B} is defined as the following formal power series over the variables $s_1, s_2, \dots, t_1, t_2, \dots$:

$$Z_{\mathcal{B}}(s_1, s_2, \dots; t_1, t_2, \dots) = \sum_{(G, c) \in \mathcal{B}} \sum_{(G^\ell, c^\ell, \sigma, v) \in \text{RSym}((G, c))} \frac{1}{|G|!} t_{|c^\ell|} \prod_{i=1}^{|G|} s_i^{c'_i(\sigma)}$$

where $c'_i(\sigma)$ is defined as the number of cycles in σ of length i that are not equal to c^ℓ (in a sense, the marked cycle c^ℓ is moved outside of the inner product and accounted for with a t_i variable instead of an s_i variable).

Cycle index sums commute with sums and products of combinatorial classes, in the sense that

$$Z_{\mathcal{B} + \mathcal{C}} = Z_{\mathcal{B}} + Z_{\mathcal{C}} \quad \text{and} \quad Z_{\mathcal{B} \times \mathcal{C}} = Z_{\mathcal{B}} \cdot Z_{\mathcal{C}}.$$

Furthermore, the cycle index sum is a generalization of its ordinary generating function, in the sense that replacing s_i with z^i and t_i with z^i in a cycle index sum results in the OGF of the class.

Lemma 4.16. If \mathcal{B} is a non cycle-pointed class, $Z_{\mathcal{B}}(z, z^2, \dots) = \mathcal{B}(z)$. Similarly, if \mathcal{B} is a cycle-pointed class, $Z_{\mathcal{B}}(z, z^2, \dots; z, z^2, \dots) = \mathcal{B}(z)$.

Proof. First consider the case when \mathcal{B} is non cycle-pointed. Then

$$\begin{aligned} Z_{\mathcal{B}}(z, z^2, \dots) &= \sum_{G \in \mathcal{B}} \sum_{(G^\ell, \sigma) \in \text{Sym}(G)} \frac{1}{|G|!} \prod_{i=1}^{|G|} z^{ic_i(\sigma)} \\ &= \sum_{G \in \mathcal{B}} \sum_{(G^\ell, \sigma) \in \text{Sym}(G)} \frac{1}{|G|!} z^{\sum_{i=1}^{|G|} ic_i(\sigma)} \\ &= \sum_{G \in \mathcal{B}} \sum_{(G^\ell, \sigma) \in \text{Sym}(G)} \frac{1}{|G|!} z^{|G|}. \end{aligned}$$

By Lemma 4.7, G has $|G|!$ symmetries, so this expression reduces to

$$\sum_{G \in \mathcal{B}} z^{|G|} = \mathcal{B}(z).$$

In the case when \mathcal{B} is cycle-pointed, we have

$$\begin{aligned}
Z_{\mathcal{B}}(z, z^2, \dots; z, z^2, \dots) &= \sum_{(G, c) \in \mathcal{B}} \sum_{(G^\ell, c^\ell, \sigma, v) \in \text{RSym}((G, c))} \frac{1}{|G|!} z^{|c^\ell|} \prod_{i=1}^{|G|} z^{i c'_i(\sigma)} \\
&= \sum_{(G, c) \in \mathcal{B}} \sum_{(G^\ell, c^\ell, \sigma, v) \in \text{RSym}((G, c))} \frac{1}{|G|!} z^{|c^\ell| + \sum_{i=1}^{|G|} i c'_i(\sigma)} \\
&= \sum_{(G, c) \in \mathcal{B}} \sum_{(G^\ell, c^\ell, \sigma, v) \in \text{RSym}((G, c))} \frac{1}{|G|!} z^{|G|}.
\end{aligned}$$

By Lemma 4.9, (G, c) has $|G|!$ rooted c -symmetries, so this expression reduces to

$$\sum_{(G, c) \in \mathcal{B}} z^{|G|} = \mathcal{B}(z).$$

□

The cycle index sums for some common classes are shown in Table 3 (we refer to Bodirsky *et al.* [19] for proof). In addition, the cycle index sums for size-restricted SET, SEQ, and CYC classes and their

Class	Cycle index sum
ε	$Z_\varepsilon = 0$
\mathcal{Z}	$Z_{\mathcal{Z}} = s_1$
SET	$Z_{\text{SET}} = \exp\left(\sum_{i=1}^{\infty} \frac{s_i}{i}\right)$
SEQ	$Z_{\text{SEQ}} = \frac{1}{1 - s_1}$
CYC	$Z_{\text{CYC}} = 1 + \sum_{i=1}^{\infty} \frac{\phi(i)}{i} \log\left(\frac{1}{1 - s_i}\right)$
SET°	$Z_{\text{SET}^\circ} = \left(\sum_{\ell \geq 1} t_\ell\right) \cdot \exp\left(\sum_{i=1}^{\infty} \frac{s_i}{i}\right)$
$\text{SET}^{\mathfrak{S}}$	$Z_{\text{SET}^{\mathfrak{S}}} = \left(\sum_{\ell \geq 2} t_\ell\right) \cdot \exp\left(\sum_{i=1}^{\infty} \frac{s_i}{i}\right)$

Table 3. Some common cycle index sums.

cycle-pointed versions can often be easily computed from Definition 4.15. For example, SET_2 has one element $\{\bullet, \bullet\}$ with two symmetries, the identity σ_{id} and the permutation σ_{sw} that swaps the two atoms, so

$$Z_{\text{SET}_2} = \frac{1}{2!} \prod_{i=1}^2 s_i^{c_i(\sigma_{\text{id}})} + \frac{1}{2!} \prod_{i=1}^2 s_i^{c_i(\sigma_{\text{sw}})} = \frac{1}{2} s_1^2 + \frac{1}{2} s_2.$$

Similarly, SET_2° has two elements each with two rooted c -symmetries, where the first element has a marked cycle of length 1 and the second has a marked cycle of length 2, so its cycle index sum is

$$Z_{\text{SET}_2^\circ} = 2 \cdot \frac{1}{2!} t_1 \prod_{i=1}^2 s_i^{c'_i(\sigma_{\text{id}})} + 2 \cdot \frac{1}{2!} t_2 \prod_{i=1}^2 s_i^{c'_i(\sigma_{\text{sw}})} = s_1 t_1 + t_2.$$

Since $\text{SET}_2^\mathcal{S}$ excludes the element of SET_2° whose marked cycle has length 1, its cycle index sum is simply $Z_{\text{SET}_2^\mathcal{S}} = t_2$.

The power of cycle index sums is that they provide us with transfer theorems for the substitution and cycle-pointed substitution operators, as shown in Table 4 [18, 19]. With these rules, we can

Class	OGF
$\mathcal{B} + \mathcal{C}$	$\mathcal{B}(z) + \mathcal{C}(z)$
$\mathcal{B} \times \mathcal{C}$	$\mathcal{B}(z) \cdot \mathcal{C}(z)$
$\mathcal{B}(\mathcal{C})$	$Z_{\mathcal{B}}(\mathcal{C}(z), \mathcal{C}(z^2), \mathcal{C}(z^3), \dots)$
$\mathcal{B} \odot \mathcal{C}$	$Z_{\mathcal{B}}(\mathcal{C}(z), \mathcal{C}(z^2), \dots; z\mathcal{C}'(z), z^2\mathcal{C}'(z^2), \dots)$

Table 4. Operator transfer theorems for unlabeled classes.

use the following procedure to develop a system of equations for $\mathcal{A}^\circ(z)$ and thereby determine the enumeration of \mathcal{A} :

1. Develop a combinatorial specification for \mathcal{A}° (*cf.* Section 4.2.1).
2. For each class that appears as the first argument of a substitution or cycle-pointed substitution, determine its cycle index sum.
3. For all other classes that appear in the specification, determine their ordinary generating function.
4. Apply the rules in Table 4 to translate the specification to a system of generating function equations.
5. Compute the coefficients \mathcal{A}_n° and $\mathcal{A}_n = \frac{1}{n} \mathcal{A}_n^\circ$ from this system of equations. A tool such as `combstruct` is often helpful at this step in order to determine the coefficients of recursively specified generating functions.

4.3.2 Example

Recall the decomposition of the class \mathcal{T}° from Section 4.2.2:

$$\begin{aligned} \mathcal{T}^\circ &= \mathcal{Z}^\bullet \times \text{SET}_{1,3,4}(\mathcal{S}) + \mathcal{Z} \times (\text{SET}_{3,4}^\mathcal{S} \odot \mathcal{S}) + \text{SET}_2^\mathcal{S} \odot \mathcal{S} \\ \mathcal{S} &= \mathcal{Z} + \mathcal{Z} \times \text{SET}_{2,3}(\mathcal{S}). \end{aligned}$$

The cycle index sums for $\text{SET}_{1,3,4}$, $\text{SET}_{2,3}$, $\text{SET}_2^{\mathcal{G}}$, and $\text{SET}_{3,4}^{\mathcal{G}}$ are¹

$$\begin{aligned} Z_{\text{SET}_{1,3,4}} &= s_1 + \frac{1}{6}s_1^3 + \frac{1}{2}s_1s_2 + \frac{1}{3}s_3 + \frac{1}{24}s_1^4 + \frac{1}{4}s_1^2s_2 + \frac{1}{8}s_2^2 + \frac{1}{3}s_1s_3 + \frac{1}{4}s_4 \\ Z_{\text{SET}_{2,3}} &= \frac{1}{2}s_1^2 + \frac{1}{2}s_2 + \frac{1}{6}s_1^3 + \frac{1}{2}s_1s_2 + \frac{1}{3}s_3 \\ Z_{\text{SET}_2^{\mathcal{G}}} &= t_2 \\ Z_{\text{SET}_{3,4}^{\mathcal{G}}} &= t_2s_1 + t_3 + \frac{1}{2}t_2s_1^2 + \frac{1}{2}t_2s_2 + t_3s_1 + t_4 \end{aligned}$$

so by Table 4 we have the following system of ordinary generating function equations:

$$\begin{aligned} \mathcal{T}^{\circ}(z) &= z \cdot \left[\mathcal{S}(z) + \frac{1}{6}\mathcal{S}(z)^3 + \frac{1}{2}\mathcal{S}(z)\mathcal{S}(z^2) + \frac{1}{3}\mathcal{S}(z^3) + \frac{1}{24}\mathcal{S}(z)^4 + \frac{1}{4}\mathcal{S}(z)^2\mathcal{S}(z^2) + \frac{1}{8}\mathcal{S}(z^2)^2 + \frac{1}{3}\mathcal{S}(z)\mathcal{S}(z^3) + \frac{1}{4}\mathcal{S}(z^4) \right] \\ &\quad + z \cdot \left[z^2\mathcal{S}'(z^2)\mathcal{S}(z) + z^3\mathcal{S}'(z^3) + \frac{1}{2}z^2\mathcal{S}'(z^2)\mathcal{S}(z)^2 + \frac{1}{2}z^2\mathcal{S}'(z^2)\mathcal{S}(z^2) + z^3\mathcal{S}'(z^3)\mathcal{S}(z) + z^4\mathcal{S}'(z^4) \right] \\ &\quad + z^2\mathcal{S}'(z^2) \\ \mathcal{S}(z) &= z + z \cdot \left[\frac{1}{2}\mathcal{S}(z)^2 + \frac{1}{2}\mathcal{S}(z^2) + \frac{1}{6}\mathcal{S}(z)^3 + \frac{1}{2}\mathcal{S}(z)\mathcal{S}(z^2) + \frac{1}{3}\mathcal{S}(z^3) \right] \end{aligned}$$

Using `comstruct`, we can compute $\mathcal{S}(z)$ to arbitrary accuracy:

$$\mathcal{S}(z) = z + z^3 + z^4 + z^5 + 2z^6 + 3z^7 + 5z^8 + 8z^9 + 14z^{10} + 23z^{11} + 40z^{12} + 70z^{13} + 122z^{14} + 217z^{15} + \dots,$$

and from the expression for $\mathcal{T}^{\circ}(z)$, by differentiating $\mathcal{S}(z)$ and substituting $z \rightarrow z^i$ as appropriate, we can compute $\mathcal{T}^{\circ}(z)$ to arbitrary accuracy:

$$\mathcal{T}^{\circ}(z) = 2z^2 + 4z^4 + 5z^5 + 6z^6 + 7z^7 + 16z^8 + 18z^9 + 40z^{10} + 55z^{11} + 96z^{12} + 156z^{13} + 280z^{14} + 435z^{15} + \dots$$

Finally, from Corollary 4.11 we have

$$\mathcal{T}(z) = \int \frac{\mathcal{T}^{\circ}(z)}{z} dz = z^2 + z^4 + z^5 + z^6 + z^7 + 2z^8 + 2z^9 + 4z^{10} + 5z^{11} + 8z^{12} + 12z^{13} + 20z^{14} + 29z^{15} + \dots$$

We note that this agrees with the enumeration of \mathcal{A} computed in Section 3.3.

¹For example, $Z_{\text{SET}_{2,3}}$ can be derived as follows. The single element of SET_3 has six symmetries: the identity, which contributes a s_1^3 term (one s_1 for each 1-cycle) to the cycle index sum; three symmetries that swap a pair of vertices, each of which contributes an s_1s_2 term (for the 1-cycle and the 2-cycle); and two 3-cycle symmetries, each of which contributes an s_3 term. Summing these and dividing by $3!$ gives the expression $Z_{\text{SET}_3} = \frac{1}{6}s_1^3 + \frac{1}{2}s_1s_2 + \frac{1}{3}s_3$, and adding this to the expression for Z_{SET_2} from Section 4.3.1 gives the stated expression for $Z_{\text{SET}_{2,3}}$.

As a second example, $Z_{\text{SET}_{3,4}^{\mathcal{G}}}$ can be derived as follows. $\text{SET}_3^{\mathcal{G}}$ has two elements; one with a marked 2-cycle, and the other with a marked 3-cycle. The first has six rooted c -symmetries, corresponding to the three possible labelings and two possible rootings on the marked cycle, and each contributes a t_2s_1 term to the cycle index sum (for the marked 2-cycle and the unmarked 1-cycle). The second also has six rooted c -symmetries, corresponding to the two possible labelings and three possible rootings on the marked cycle, and each contributes a t_3 term. Summing these and dividing by $3!$ gives the expression $Z_{\text{SET}_3^{\mathcal{G}}} = t_2s_1 + t_3$.

In addition, $\text{SET}_4^{\mathcal{G}}$ has three elements; one with a marked 2-cycle, one with a marked 3-cycle, and one with a marked 4-cycle. The first has 24 rooted c -symmetries, 12 of which have an unmarked 2-cycle and contribute a t_2s_2 term to the cycle index sum, and the other 12 of which have two unmarked 1-cycles and contribute a $t_2s_1^2$ term. The second also has 24 rooted c -symmetries, corresponding to the eight possible labelings and three possible rootings on the marked cycle, and each contributes a t_3s_1 term (for the marked 3-cycle and the unmarked 1-cycle). The third also has 24 rooted c -symmetries, corresponding to the six possible labelings and four possible rootings on the marked cycle, and each contributes a t_4 term. Summing these and dividing by $4!$ gives the expression $Z_{\text{SET}_4^{\mathcal{G}}} = \frac{1}{2}t_2s_1^2 + \frac{1}{2}t_2s_2 + t_3s_1 + t_4$. Finally, summing the expressions for $Z_{\text{SET}_3^{\mathcal{G}}}$ and $Z_{\text{SET}_4^{\mathcal{G}}}$ results in the stated expression for $Z_{\text{SET}_{3,4}^{\mathcal{G}}}$.

4.4 Boltzmann samplers

4.4.1 Theory

In this section we describe how to build a Boltzmann sampler for \mathcal{A}° from its combinatorial specification, which (by Corollary 4.11) then provides an unbiased sampler for \mathcal{A} . To accomplish this we use *Pólya-Boltzmann samplers*, which generalize Boltzmann samplers in the same manner that cycle index sums generalize ordinary generating functions.

Definition 4.17. Suppose that \mathcal{B} is a non cycle-pointed class. For fixed parameters (s_1, s_2, \dots) at which $Z_{\mathcal{B}}(s_1, s_2, \dots)$ converges, a *Pólya-Boltzmann sampler* for \mathcal{B} is a random generator $\Gamma Z_{\mathcal{B}}(s_1, s_2, \dots)$ that draws an object (G^ℓ, σ) from $\bigcup_{G \in \mathcal{B}} \text{Sym}(G)$ with probability

$$\mathbb{P}_{(s_1, s_2, \dots)}[G^\ell, \sigma] = \frac{1}{Z_{\mathcal{B}}(s_1, s_2, \dots)} \frac{1}{|G^\ell|!} \prod_{i=1}^{|G^\ell|} s_i^{c_i(\sigma)}.$$

If \mathcal{B} is a cycle-pointed class, then for fixed parameters $(s_1, s_2, \dots; t_1, t_2, \dots)$ at which $Z_{\mathcal{B}}(s_1, s_2, \dots; t_1, t_2, \dots)$ converges, a *Pólya-Boltzmann sampler* for \mathcal{B} is a random generator $\Gamma Z_{\mathcal{B}}(s_1, s_2, \dots; t_1, t_2, \dots)$ that draws an object $(G^\ell, c^\ell, \sigma, v)$ from $\bigcup_{(G, c) \in \mathcal{B}} \text{RSym}((G, c))$ with probability

$$\mathbb{P}_{(s_1, s_2, \dots; t_1, t_2, \dots)}[G^\ell, c^\ell, \sigma, v] = \frac{1}{Z_{\mathcal{B}}(s_1, s_2, \dots; t_1, t_2, \dots)} \frac{1}{|G^\ell|! |t|_{c^\ell}} \prod_{i=1}^{|G^\ell|} s_i^{c'_i(\sigma)}.$$

A Pólya-Boltzmann sampler for a class \mathcal{B} is a generalization of the ordinary Boltzmann sampler, in the sense that sampling at the parameters $s_i = z^i$ and $t_i = z^i$ and returning the underlying structure of the symmetry/rooted c -symmetry results in a Boltzmann sampler for \mathcal{B} .

Lemma 4.18. *If \mathcal{B} is a non cycle-pointed class,*

$$\Gamma \mathcal{B}(z) = ((G^\ell, \sigma) \leftarrow \Gamma Z_{\mathcal{B}}(z, z^2, \dots); \text{ \textbf{return} } G)$$

is a Boltzmann sampler for \mathcal{B} . Similarly, if \mathcal{B} is a cycle-pointed class,

$$\Gamma \mathcal{B}(z) = ((G^\ell, c^\ell, \sigma, v) \leftarrow \Gamma Z_{\mathcal{B}}(z, z^2, \dots; z, z^2, \dots); \text{ \textbf{return} } (G, c))$$

is a Boltzmann sampler for \mathcal{B} .

Proof. If \mathcal{B} is not cycle-pointed, then by Lemma 4.16 the probability of returning $(G^\ell, \sigma) \in \text{Sym}(G)$ from $\Gamma \mathcal{B}(z)$ is

$$\frac{1}{\mathcal{B}(z)} \frac{1}{|G|!} z^{\sum_{i=1}^{|G|} i c_i(\sigma)} = \frac{1}{\mathcal{B}(z)} \frac{1}{|G|!} z^{|G|}.$$

Since G has $|G|!$ symmetries (*cf.* Lemma 4.7), the probability of returning one such symmetry from $\Gamma Z_{\mathcal{B}}(z, z^2, \dots)$, and hence of returning G from $\Gamma \mathcal{B}(z)$, is

$$\frac{z^{|G|}}{\mathcal{B}(z)}.$$

A parallel argument holds in the case when \mathcal{B} is cycle-pointed, by using Lemma 4.9 in place of Lemma 4.7. \square

<pre> $\Gamma Z_{\text{SET}}(s_1, s_2, \dots)$: $K \leftarrow \text{MAX_INDEX}(s_1, s_2, \dots)$ for k from 1 to $K - 1$ do $i_k \leftarrow \text{Pois}\left(\frac{s_k}{k}\right)$ end for $i_K \leftarrow \text{Pois}_{\geq 1}\left(\frac{s_K}{K}\right)$ Let S be a set of $\sum_{k=1}^K k i_k$ atoms. Let σ be a permutation on the atoms of S, with i_k cycles of length k for each $1 \leq k \leq K$. Randomly label the atoms of S to obtain S^ℓ. return (S^ℓ, σ) </pre>
<pre> $\Gamma Z_{\text{SET}^\circ}(s_1, s_2, \dots; t_1, t_2, \dots)$: $K \leftarrow \text{ROOT_CYCLE_SIZE}(t_1, t_2, \dots)$ $(S, \sigma) \leftarrow \Gamma Z_{\text{SET}}(s_1, s_2, \dots)$ Let S' be S with K new atoms, let c be a cycle on these K atoms, let σ' be σ together with the cycle c, and let v be an atom on c. Randomly label the atoms of S' to obtain S'^ℓ and c^ℓ. return $(S'^\ell, c^\ell, \sigma', v)$ </pre>

Figure 18. Pólya-Boltzmann samplers for SET and SET[◊].²

Pólya-Boltzmann samplers for various common classes are described by Bodirsky *et al.* [19]. We do not list them here for the sake of brevity, but for the purpose of example we include the Pólya-Boltzmann samplers for SET and SET[◊] in Figure 18.

Just as cycle index sums provide transfer theorems for the substitution and cycle-pointed substitution operators, Pólya-Boltzmann samplers allow us to build Boltzmann samplers for classes specified with these operators. To compute a Boltzmann sampler for a class specified by a substitution or cycle-pointed substitution, it suffices to have a Pólya-Boltzmann sampler for the class that appears as the first argument of the substitution and an ordinary Boltzmann sampler for the class that appears as the second argument. The corresponding algorithmic rules are shown in Table 5.

With these rules, we can use the following procedure to develop a Boltzmann sampler for $\mathcal{A}^\circ(z)$ and thereby obtain an unbiased sampler for \mathcal{A} :

1. Develop a combinatorial specification for \mathcal{A}° (*cf.* Section 4.2.1).

²MAX_INDEX(s_1, s_2, \dots) is a random generator over the integers $k \geq 1$ for the distribution

$$\mathbb{P}[K \leq k] = \frac{1}{Z_{\text{SET}}(s_1, s_2, \dots)} \prod_{i \leq k} \exp\left(\frac{s_i}{i}\right),$$

and ROOT_CYCLE_SIZE(t_1, t_2, \dots) is a random generator over the integers $k \geq 1$ for the distribution

$$\mathbb{P}[K = k] = \frac{t_k}{\sum_{i=1}^{\infty} t_i}.$$

In practice, these distributions are drawn from by inversion sampling. The infinite sums present in the probability expressions (recall that Z_{SET} has an infinite sum) can be computed to arbitrary precision using a tool such as Maple, assuming that the values of s_i and t_i are explicit functions of i .

Class	Boltzmann sampler
$\mathcal{B} + \mathcal{C}$	if $\text{Bern}\left(\frac{\mathcal{B}(z)}{\mathcal{B}(z) + \mathcal{C}(z)}\right)$ then $\Gamma\mathcal{B}(z)$ else $\Gamma\mathcal{C}(z)$
$\mathcal{B} \times \mathcal{C}$	$(\Gamma\mathcal{B}(z), \Gamma\mathcal{C}(z))$
$\mathcal{B}(\mathcal{C})$	$(B, \sigma) \leftarrow \Gamma Z_{\mathcal{B}}(\mathcal{C}(z), \mathcal{C}(z^2), \dots)$ for each cycle c of σ do $C \leftarrow \Gamma\mathcal{C}(z^{ c })$ Replace each atom of c by a copy of C end for return the resulting structure
$\mathcal{B} \odot \mathcal{C}$	$((B, c), \sigma, v) \leftarrow \Gamma Z_{\mathcal{B}}(\mathcal{C}(z), \mathcal{C}(z^2), \dots; z\mathcal{C}'(z), z^2\mathcal{C}'(z^2), \dots)$ for each unmarked cycle c' of σ do $C \leftarrow \Gamma\mathcal{C}(z^{ c' })$ Replace each atom of c' by a copy of C end for $(C, \tilde{c}) \leftarrow \Gamma\mathcal{C}^{\odot}(z^{ c })$ Replace each atom of c by a copy of (C, \tilde{c}) Mark the cycle that is the composition of all $ c $ copies of \tilde{c} return the resulting cycle-pointed structure

Table 5. Boltzmann sampler constructions for operators on unlabeled classes.

2. For each class that appears as the first argument of a substitution or cycle-pointed substitution, compute a Pólya-Boltzmann sampler for it.
3. For all other classes that appear in the specification, compute an ordinary Boltzmann sampler.
4. Apply the rules in Table 5 to translate the specification to a Boltzmann sampler for \mathcal{A}^{\odot} .
5. Apply Corollary 4.11 to obtain an unbiased sampler for \mathcal{A} .

4.4.2 Example

Again, recall the decomposition of the class \mathcal{T}^{\odot} from Section 4.2.2:

$$\mathcal{T}^{\odot} = \mathcal{Z}^{\bullet} \times \text{SET}_{1,3,4}(\mathcal{S}) + \mathcal{Z} \times (\text{SET}_{3,4}^{\odot} \odot \mathcal{S}) + \text{SET}_2^{\odot} \odot \mathcal{S}$$

$$\mathcal{S} = \mathcal{Z} + \mathcal{Z} \times \text{SET}_{2,3}(\mathcal{S}).$$

From the Pólya-Boltzmann samplers for SET_k and SET_k^{\odot} given by Bodirsky *et al.* [19], we may apply the substitution rules in Table 5 to derive Boltzmann samplers for $\text{SET}_k(\mathcal{A})$, $\text{SET}_k^{\odot} \odot \mathcal{A}$, and $\text{SET}_k^{\odot} \odot \mathcal{A}$ (for arbitrary k and \mathcal{A}). By combining these with the sum and product rules in Table 5, we may compute Boltzmann samplers for \mathcal{S} , \mathcal{S}^{\odot} , and \mathcal{T}^{\odot} . By Corollary 4.11, running $\Gamma\mathcal{T}^{\odot}(z)$ and forgetting the marked cycle provides an unbiased sampler for the class \mathcal{T} . The expected size of an object drawn from this sampler when the parameter is taken at the singularity is 39.710, compared

to the corresponding value of 4.224 for a Boltzmann sampler of \mathcal{T} (*cf.* Section 3.3). We provide pseudocode for these samplers in Appendix B.

5 Two interesting classes of graphs

In this section, we apply the techniques of the previous sections to analyze two interesting classes of unlabeled graphs – distance-hereditary graphs, and three-leaf power graphs. These were previously studied from an analytic combinatorics point of view by Chauve *et al.* [6], where the authors develop grammars for these classes of graphs using the dissymmetry theorem and then derive exact enumerations and asymptotics from these grammars (we review this work in Section 5.4). We aim to extend this result by analyzing the same graph classes using cycle pointing, with which we obtain enumerations that match the ones provided by Chauve *et al.* and build unbiased samplers for these two classes of graphs.

5.1 Distance-hereditary graphs

Definition 5.1. For a graph G and vertices $u, v \in V(G)$, the *distance* between u and v – denoted $d_G(u, v)$ – is defined as the length of the shortest path between u and v (or ∞ if u and v are in different connected components of G).

Definition 5.2. A subgraph $H \leq G$ is said to be *induced* if, for any $u, v \in V(H)$ such that $uv \in E(G)$, $uv \in E(H)$ as well.

Definition 5.3. [13] A connected graph G is said to be *distance-hereditary* if, for any connected induced subgraph $H \leq G$ and any vertices $u, v \in V(H)$, $d_H(u, v) = d_G(u, v)$.

Distance-hereditary graphs were first introduced in 1977 by Howorka [7], and many properties of these graphs have been discovered since then. For example, various optimization problems that are NP-hard in the general case, such as finding a Hamiltonian cycle or a maximal clique, can be solved in polynomial time for distance-hereditary graphs [15, 32, 26]. One alternate characterization that is particularly useful for recognizing a distance-hereditary graph is the following:

Lemma 5.4. *A connected graph G is distance-hereditary iff every cycle of length five or higher has at least one pair of crossing diagonals.*

5.2 Three-leaf power graphs

Definition 5.5. [14] For an integer $k > 0$, a connected graph G is said to be a *k-leaf power* if there exists a tree T with $V(T) = V(G)$ such that for any vertices $u, v \in V(G)$, $uv \in E(G)$ iff $d_T(u, v) \leq k$.

k -leaf power graphs were first introduced in 2002 by Nishimura *et al.* [24], who were interested in building *phylogenetic trees* that reconstruct the evolutionary history of a set of species or genes. We study the particular case $k = 3$, called three-leaf power graphs.

5.3 Split decomposition

Recently, various techniques have emerged for analyzing classes of graphs by using bijective representations of these graphs in terms of trees – see, for example, the correspondence between binary trees and edge-rooted 3-connected planar graphs used by Fusy [9], or the correspondence between Apollonian networks and k -trees used by Darrasse [3]. In order to study distance-hereditary and three-leaf power graphs, we employ the representation of these classes developed by Gioan and Paul [8], who use the technique of *split decomposition* to characterize distance-hereditary and three-leaf power graphs in terms of *graph-labeled trees*. We begin by giving an overview of this characterization, which we use in the following sections to enumerate and generate graphs from these classes.

Definition 5.6. A *graph-labeled tree* (T, \mathcal{F}) is a tree T in which every internal node v of degree k is labeled by a graph $G_v \in \mathcal{F}$ on k vertices, called *marker vertices*, such that there is a bijection ρ_v from the tree-edges of T incident to v to the marker vertices of G_v .

For example, in Figure 19 the internal nodes of T are denoted with large circles, the marker vertices are denoted with small hollow circles, the leaves of T are denoted with small solid circles, and the bijection ρ_v is denoted by each edge that crosses the boundary of an internal node and ends at a marker vertex. (Note: the node labels are only for convenience in discussing the vertices - the tree itself is unlabeled.)

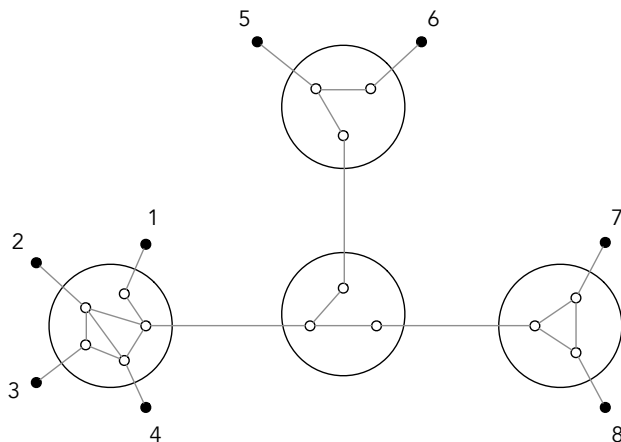


Figure 19. A graph-labeled tree.

Definition 5.7. Let (T, \mathcal{F}) be a graph-labeled tree and let $\ell, \ell' \in V(T)$ be leaves of T . We say that ℓ and ℓ' are *accessible* (or equivalently, ℓ is *accessible* from ℓ') if there exists a path from ℓ to ℓ' in T such that for any adjacent edges $e = uv$ and $e' = vw$ on the path, $\rho_v(e)\rho_v(e') \in E(G_v)$.

Informally, ℓ and ℓ' are accessible if it is possible to draw a path through the graph-labeled tree from ℓ to ℓ' that uses at most one interior edge from each graph label G_v . For example, in Figure 19, leaf 1 is accessible from leaves 5, 7, and 8, leaf 2 is accessible from leaves 3, 4, 5, 7, and 8, and leaf 3 is only accessible from leaves 2 and 4.

Definition 5.8. The *original graph* (called the *accessibility graph* by Gioan and Paul [8]) of a graph-labeled tree (T, \mathcal{F}) is the graph $G = Gr(T, \mathcal{F})$ where $V(G)$ is the leaf set of T and, for $x, y \in V(G)$, $xy \in E(G)$ iff x and y are accessible in (T, \mathcal{F}) .

For example, Figure 20 shows the original graph for the graph-labeled tree in Figure 19. We now define the *split tree* of a connected graph G , which is a particular graph-labeled tree whose original graph is G .

Definition 5.9. For $V \subseteq V(G)$, the *neighborhood* of V , denoted $N(V)$, is defined as the set of vertices in $V(G) \setminus V$ that are adjacent to at least one vertex in V .

Definition 5.10. A *split* of a graph G is a bipartition (V_1, V_2) of $V(G)$ such that

1. $|V_1| \geq 2$ and $|V_2| \geq 2$, and
2. every vertex of $N(V_1)$ is adjacent to every vertex of $N(V_2)$.

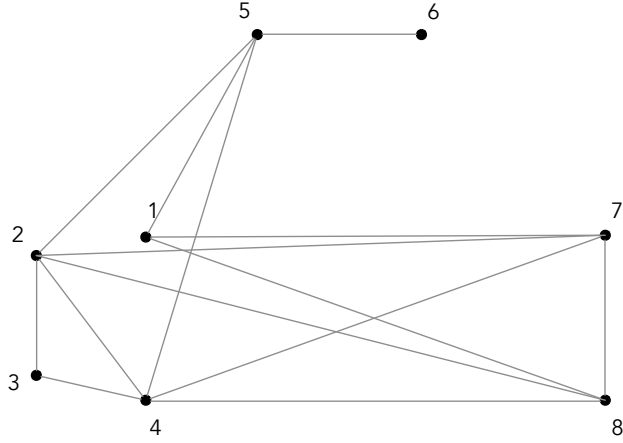


Figure 20. Original graph for the graph-labeled tree in Figure 19.

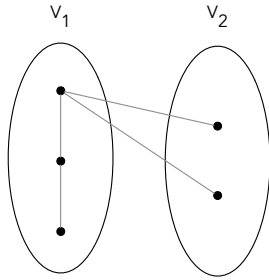


Figure 21. Valid split.

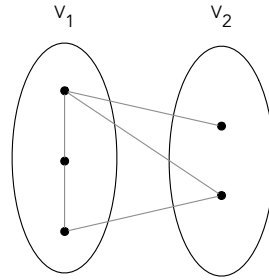


Figure 22. Invalid split.

For example, Figure 21 is a split, while Figure 22 is not a split because the bottom vertex of V_1 is in $N(V_2)$ and the top vertex of V_2 is in $N(V_1)$, but these vertices are not adjacent.

Definition 5.11. A graph G is called *prime* if it has no split, and *degenerate* if every partition of $V(G)$ into two sets of size ≥ 2 is a split. It is known that the only degenerate graphs are the cliques K_n and the stars $K_{1,n}$ for $n \geq 0$.

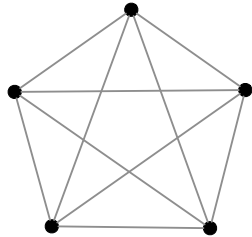


Figure 23. Clique K_5 .

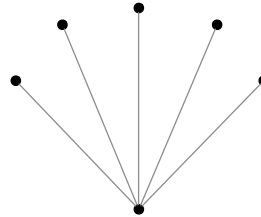


Figure 24. Star $K_{1,5}$.

Definition 5.12. A graph-labeled tree (T, \mathcal{F}) is called *reduced* if

1. every $v \in V(T)$ has degree ≥ 3 ,
2. there does not exist $uv \in E(T)$ such that G_u and G_v are both cliques, and
3. there does not exist $e = uv \in E(T)$ such that G_u and G_v are both stars, $\rho_u(e)$ is the center of G_u , and $\rho_v(e)$ is an extremity of G_v .

The intuition behind this definition is the following: if two clique nodes of sizes m and n are adjacent in (T, \mathcal{F}) , then they can be replaced with a single clique node of size $m + n - 2$ in such a way that the original graph of (T, \mathcal{F}) does not change (see Figure 25). A similar reduction can be performed when the center of one star node is adjacent to an extremity of another star node (see Figure 26).

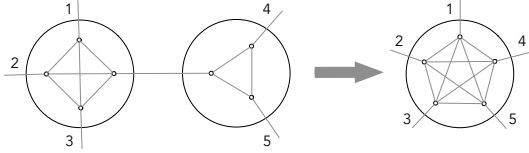


Figure 25. Reduction of two adjacent clique nodes in a split tree to a single clique (recreated from [8]).

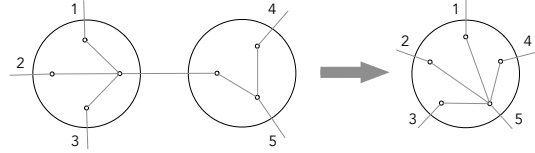


Figure 26. Reduction of two adjacent star nodes in a split tree to a single star (recreated from [8]).

The following characterization of connected graphs is shown by Gioan and Paul [8]:

Theorem 5.13. *For any connected graph G , there exists a unique reduced graph-labeled tree (T, \mathcal{F}) such that $G = Gr(T, \mathcal{F})$ and every node label $G_v \in \mathcal{F}$ is either prime or degenerate. We call this graph-labeled tree the split tree of G , and denote it by $ST(G)$.*

Gioan and Paul further show that distance-hereditary and three-leaf power graphs can be characterized by the following conditions on their split trees:

Theorem 5.14. *A graph G is distance-hereditary iff its split tree has only clique and star nodes (i.e. has only degenerate nodes). For this reason, distance-hereditary graphs are called totally decomposable with respect to the split decomposition.*

Figure 27 shows a split tree of a distance-hereditary graph.

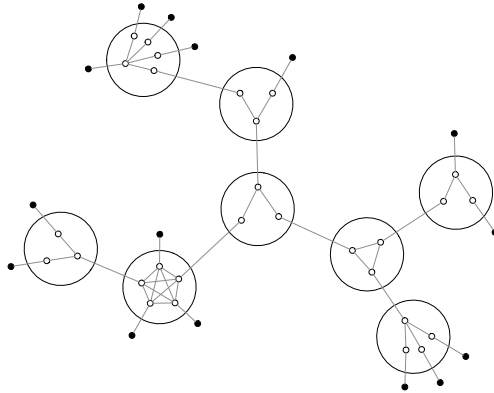


Figure 27. Split tree of a distance-hereditary graph.

Theorem 5.15. *A graph G is a three-leaf power iff its split tree $ST(G) = (T, \mathcal{F})$*

1. has only clique and star nodes,
2. the set of star nodes forms a subtree of T , and
3. the center of every star node is adjacent to either a clique node or a leaf.

We note that by condition 1, all three-leaf power graphs are distance-hereditary. Figure 28 shows a split tree of a three-leaf power graph.

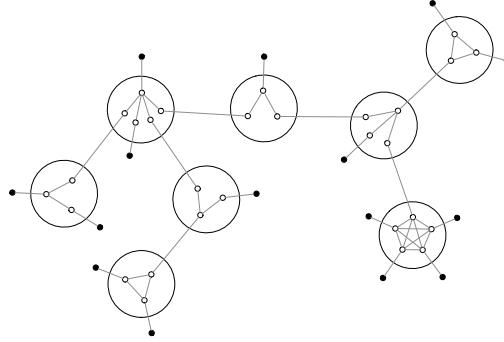


Figure 28. Split tree of a three-leaf power graph.

5.4 Enumeration with the dissymmetry theorem

Chauve *et al.* [6] use the dissymmetry theorem and the split decomposition characterization of distance-hereditary and three-leaf power graphs described in Theorems 5.14 and 5.15 to construct combinatorial grammars and enumerations for these classes. We give an overview of these results here, and make use of the same grammars in our cycle pointing analysis in Section 5.5.

5.4.1 Distance-hereditary graphs

From Theorem 5.14, Theorem 3.1, and Lemma 3.3, Chauve *et al.* [6] derive the following grammar for the class \mathcal{DH} of split trees of distance-hereditary graphs:

$$\begin{aligned}
\mathcal{DH} + \mathcal{T}_{\mathcal{K}-s} + \mathcal{T}_{s \rightarrow s} &\simeq \mathcal{T}_{\mathcal{K}} + \mathcal{T}_s + \mathcal{T}_{s-s} \\
\mathcal{T}_{\mathcal{K}} &= \text{SET}_{\geq 3}(\mathcal{Z} + \mathcal{S}_{\mathcal{X}} + \mathcal{S}_{\mathcal{C}}) \\
\mathcal{T}_s &= \mathcal{S}_{\mathcal{C}} \times (\mathcal{Z} + \mathcal{K} + \mathcal{S}_{\mathcal{C}}) \\
\mathcal{T}_{\mathcal{K}-s} &= \mathcal{K} \times (\mathcal{S}_{\mathcal{C}} + \mathcal{S}_{\mathcal{X}}) \\
\mathcal{T}_{s-s} &= \text{SET}_2(\mathcal{S}_{\mathcal{C}}) + \text{SET}_2(\mathcal{S}_{\mathcal{X}}) \\
\mathcal{T}_{s \rightarrow s} &= \mathcal{S}_{\mathcal{C}} \times \mathcal{S}_{\mathcal{C}} + \mathcal{S}_{\mathcal{X}} \times \mathcal{S}_{\mathcal{X}} \\
\mathcal{K} &= \text{SET}_{\geq 2}(\mathcal{Z} + \mathcal{S}_{\mathcal{C}} + \mathcal{S}_{\mathcal{X}}) \\
\mathcal{S}_{\mathcal{C}} &= \text{SET}_{\geq 2}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_{\mathcal{X}}) \\
\mathcal{S}_{\mathcal{X}} &= (\mathcal{Z} + \mathcal{K} + \mathcal{S}_{\mathcal{C}}) \times \text{SET}_{\geq 1}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_{\mathcal{X}})
\end{aligned}$$

Explanation of the classes.

- \mathcal{K} : A clique node with one of its incident subtrees having been removed

- \mathcal{S}_c : A star node with the subtree incident to its center having been removed
- \mathcal{S}_x : A star node with the subtree incident to one of its extremities having been removed.
- \mathcal{T}_K : A distance-hereditary split tree rooted at a clique node
- \mathcal{T}_s : A distance-hereditary split tree rooted at a star node
- \mathcal{T}_{K-s} : A distance-hereditary split tree rooted at an undirected edge connecting a clique node and a star node
- \mathcal{T}_{s-s} : A distance-hereditary split tree rooted at an undirected edge connecting two star nodes
- $\mathcal{T}_{s \rightarrow s}$: A distance-hereditary split tree rooted at a directed edge connecting two star nodes

Sketch of proof. The three mutually-recursive expressions for \mathcal{K} , \mathcal{S}_c , and \mathcal{S}_x encode the requirements that a split tree must have all nodes of degree at least 3, no adjacent clique nodes, and no extremity of a star node adjacent to the center of another star node.

By Theorem 3.1 and Lemma 3.3,

$$\begin{aligned} \mathcal{DH} + \mathcal{T}_{s \rightarrow s} + \mathcal{T}_{K \rightarrow K} + \mathcal{T}_{s \rightarrow K} + \mathcal{T}_{K \rightarrow s} \\ \simeq \mathcal{T}_K + \mathcal{T}_s + \mathcal{T}_{s-s} + \mathcal{T}_{K-s} + \mathcal{T}_{K-K}. \end{aligned}$$

Noting that $\mathcal{T}_{K \rightarrow K}$ and \mathcal{T}_{K-K} are empty (since clique nodes cannot be adjacent in a split tree) and $\mathcal{T}_{s \rightarrow K} \simeq \mathcal{T}_{K \rightarrow s} \simeq \mathcal{T}_{K-s}$, it follows that

$$\mathcal{DH} + \mathcal{T}_{K-s} + \mathcal{T}_{s \rightarrow s} \simeq \mathcal{T}_K + \mathcal{T}_s + \mathcal{T}_{s-s}.$$

□

Using the Maple package `comstruct`, Chauve *et al.* compute the enumeration of each of the above classes, and hence of the class \mathcal{DH} .

Corollary 5.16. *The first few terms of the OGF of the class of distance-hereditary graphs are*

$$\mathcal{DH}(z) = z + z^2 + 2z^3 + 6z^4 + 18z^5 + 73z^6 + 308z^7 + 1484z^8 + 7492z^9 + 40010z^{10} + \dots$$

5.4.2 Three-leaf power graphs

From Theorem 5.15, Theorem 3.1, and Lemma 3.3, Chauve *et al.* [6] derive the following grammar for the class $3\mathcal{LP}$ of (split trees of) three-leaf power graphs:

$$\begin{aligned} 3\mathcal{LP} + \mathcal{T}_{s \rightarrow s} &\simeq \mathcal{K} + \mathcal{T}_s + \mathcal{T}_{s-s} \\ \mathcal{T}_s &= \mathcal{A} \times \mathcal{S}_c \\ \mathcal{T}_{s-s} &= \text{SET}_2(\mathcal{S}_x) \\ \mathcal{T}_{s \rightarrow s} &= \mathcal{S}_x \times \mathcal{S}_x \\ \mathcal{S}_c &= \text{SET}_{\geq 2}(\mathcal{A} + \mathcal{S}_x) \\ \mathcal{S}_x &= \mathcal{A} \times \text{SET}_{\geq 1}(\mathcal{A} + \mathcal{S}_x) \\ \mathcal{A} &= \mathcal{Z} + \text{SET}_{\geq 2}(\mathcal{Z}) = \text{SET}_{\geq 1}(\mathcal{Z}) \\ \mathcal{K} &= \text{SET}_{\geq 3}(\mathcal{Z}) \end{aligned}$$

Explanation of the classes.

- \mathcal{A} : The disjoint union of a leaf and a clique having been entered through one of its edges
- \mathcal{S}_c : A star node with the subtree incident to its center having been removed
- \mathcal{S}_x : A star node with the subtree incident to one of its extremities having been removed
- \mathcal{T}_s : A three-leaf power split tree rooted at a star node
- \mathcal{K} : A three-leaf power split tree consisting of a single clique node
- \mathcal{T}_{s-s} : A three-leaf power split tree rooted at an undirected edge connecting two star nodes
- $\mathcal{T}_{s \rightarrow s}$: A three-leaf power split tree rooted at a directed edge connecting two star nodes

Sketch of proof. The expressions for S_C and S_X encode the requirements that the split tree of a three-leaf power graph consists of either a single clique node or a subtree of star nodes joined at their extremities together with either a leaf or a clique node pending from each center and remaining extremity of these star nodes. Due to this structure, the leaves/clique nodes represented by \mathcal{A} can be thought of as “meta-leaves” and excluded from the dissymmetry theorem by a similar argument as the one in Lemma 3.3. Then

$$3\mathcal{LP} + \mathcal{T}_{s \rightarrow s} \simeq \mathcal{K} + \mathcal{T}_s + \mathcal{T}_{s-s}$$

follows immediately from Theorem 3.1. □

Using `comstruct`, Chauve *et al.* compute the enumeration of each of the above classes, and hence of the class $3\mathcal{LP}$.

Corollary 5.17. *The first few terms of the OGF of the class of three-leaf power graphs are*

$$3\mathcal{LP}(z) = z + z^2 + 2z^3 + 5z^4 + 12z^5 + 32z^6 + 82z^7 + 227z^8 + 629z^9 + 1840z^{10} + \dots$$

5.5 Enumeration and sampling with cycle pointing

5.5.1 Distance-hereditary graphs

In this section we apply the steps outlined in Sections 4.2, 4.3, and 4.4 to enumerate and build an unbiased sampler for the class \mathcal{DH} of distance-hereditary graphs. Instead of working directly with the graphs, we will work with their split tree decompositions.

Let \mathcal{S}_x , \mathcal{S}_c , and \mathcal{K} be defined as in Section 5.4.1. Then we have

$$\mathcal{S}_x = (\mathcal{Z} + \mathcal{K} + \mathcal{S}_c) \times \text{SET}_{\geq 1}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_x)$$

$$\mathcal{S}_c = \text{SET}_{\geq 2}(\mathcal{Z} + \mathcal{K} + \mathcal{S}_x)$$

$$\mathcal{K} = \text{SET}_{\geq 2}(\mathcal{Z} + \mathcal{S}_c + \mathcal{S}_x)$$

Since the cycle index sums of SET , SET_0 , and SET_1 are

$$Z_{\text{SET}}(s_1, s_2, \dots) = \exp\left(\sum_{i=1}^{\infty} \frac{s_i}{i}\right) \quad Z_{\text{SET}_0}(s_1, s_2, \dots) = 1 \quad Z_{\text{SET}_1}(s_1, s_2, \dots) = s_1$$

it follows that

$$Z_{\text{SET}_{\geq 1}}(s_1, s_2, \dots) = \exp\left(\sum_{i=1}^{\infty} \frac{s_i}{i}\right) - 1$$

and

$$Z_{\text{SET}_{\geq 2}}(s_1, s_2, \dots) = \exp\left(\sum_{i=1}^{\infty} \frac{s_i}{i}\right) - 1 - s_1.$$

From Table 4, we obtain the equations

$$\begin{aligned} \mathcal{S}_X(z) &= (z + \mathcal{K}(z) + \mathcal{S}_E(z)) \cdot \left[\exp\left(\sum_{i=1}^{\infty} \frac{1}{i} [z^i + \mathcal{K}(z^i) + \mathcal{S}_X(z^i)]\right) - 1 \right] \\ \mathcal{S}_E(z) &= \exp\left(\sum_{i=1}^{\infty} \frac{1}{i} [z^i + \mathcal{K}(z^i) + \mathcal{S}_X(z^i)]\right) - 1 - z - \mathcal{K}(z) - \mathcal{S}_X(z) \\ \mathcal{K}(z) &= \exp\left(\sum_{i=1}^{\infty} \frac{1}{i} [z^i + \mathcal{S}_E(z^i) + \mathcal{S}_X(z^i)]\right) - 1 - z - \mathcal{S}_E(z) - \mathcal{S}_X(z), \end{aligned}$$

and using `combstruct` we can compute the coefficients of these generating functions:

$$\begin{aligned} \mathcal{S}_X(z) &= z^2 + 5z^3 + 23z^4 + 119z^5 + 639z^6 + 3629z^7 + 21257z^8 + 127995z^9 + 786481z^{10} + \dots \\ \mathcal{S}_E(z) &= z^2 + 3z^3 + 14z^4 + 67z^5 + 367z^6 + 2065z^7 + 12150z^8 + 73177z^9 + 450322z^{10} + \dots \\ \mathcal{K}(z) &= z^2 + 3z^3 + 14z^4 + 67z^5 + 367z^6 + 2065z^7 + 12150z^8 + 73177z^9 + 450322z^{10} + \dots \end{aligned}$$

We now build a specification for \mathcal{DH}° in terms of \mathcal{S}_X , \mathcal{S}_E , and \mathcal{K} :

$$\begin{aligned} \mathcal{DH}^\circ &= \mathcal{Z}^\circ \\ &\quad + \text{SET}_2^\circ \odot \mathcal{Z} \\ &\quad + \mathcal{Z}^\circ \times (\mathcal{S}_X + \mathcal{S}_E + \mathcal{K}) \\ &\quad + \text{SET}_2^\circ \odot \mathcal{S}_X \\ &\quad + \text{SET}_2^\circ \odot \mathcal{S}_E \\ &\quad + \text{SET}_{\geq 3}^\circ \odot (\mathcal{Z} + \mathcal{S}_X + \mathcal{S}_E) \\ &\quad + (\mathcal{Z} + \mathcal{K} + \mathcal{S}_E) \times \text{SET}^\circ \odot (\mathcal{Z} + \mathcal{K} + \mathcal{S}_X) \end{aligned}$$

Term i in this specification corresponds to case i of the following cases:

1. The tree has one leaf.
2. The tree has two leaves.
3. The marked cycle has length 1 and the tree has > 2 leaves.
4. The marked cycle has length ≥ 2 and has as its center an edge connecting two star nodes at extremities.
5. The marked cycle has length ≥ 2 and has as its center an edge connecting two star nodes at their centers.
6. The marked cycle has length ≥ 2 and has as its center a clique node.
7. The marked cycle has length ≥ 2 and has as its center a star node.

In a split tree, there can be no edge connecting two clique nodes; furthermore, while there can be an edge connecting a clique node and a star node, such an edge cannot be the center of a cycle of an automorphism of the split tree. Thus these seven cases cover all possibilities for a cycle-pointed split tree of a distance-hereditary graph.

Next, we translate this specification into a generating function equation. Recall that

$$\begin{aligned} Z_{\text{SET}_2^{\mathcal{G}}}(s_1, s_2, \dots; t_1, t_2, \dots) &= t_2 \\ Z_{\text{SET}^{\mathcal{G}}}(s_1, s_2, \dots; t_1, t_2, \dots) &= \left(\sum_{l=2}^{\infty} t_l \right) \cdot \exp \left(\sum_{i=1}^{\infty} \frac{s_i}{i} \right) \\ Z_{\text{SET}_{\geq 3}^{\mathcal{G}}}(s_1, s_2, \dots; t_1, t_2, \dots) &= \left(\sum_{l=2}^{\infty} t_l \right) \cdot \exp \left(\sum_{i=1}^{\infty} \frac{s_i}{i} \right) - t_2 \end{aligned}$$

By Theorem 4.5, we know that the OGF for \mathcal{DH}° is $z \cdot \mathcal{DH}'(z)$, so by the transfer theorems from Table 4 it follows that

$$\begin{aligned} z \cdot \mathcal{DH}'(z) &= z + 2z^2 + z[\mathcal{S}_X(z) + \mathcal{S}_C(z) + \mathcal{K}(z)] + z^2\mathcal{S}_X'(z^2) + z^2\mathcal{S}_C'(z^2) \\ &\quad + \left(\sum_{l=2}^{\infty} z^l [1 + \mathcal{S}_X'(z^l) + \mathcal{S}_C'(z^l)] \right) \cdot \exp \left(\sum_{i=1}^{\infty} \frac{1}{i} [z^i + \mathcal{S}_X(z^i) + \mathcal{S}_C(z^i)] \right) - z^2(1 + \mathcal{S}_X'(z^2) + \mathcal{S}_C'(z^2)) \\ &\quad + (z + \mathcal{K}(z) + \mathcal{S}_C(z)) \left(\sum_{l=2}^{\infty} z^l [1 + \mathcal{K}'(z^l) + \mathcal{S}_X'(z^l)] \right) \cdot \exp \left(\sum_{i=1}^{\infty} \frac{1}{i} [z^i + \mathcal{K}(z^i) + \mathcal{S}_X(z^i)] \right) \end{aligned}$$

By differentiating the earlier expressions for $\mathcal{S}_X(z)$, $\mathcal{S}_C(z)$, and $\mathcal{K}(z)$ and using these to simplify the above formula, we reduce it to

$$\begin{aligned} z \cdot \mathcal{DH}'(z) &= z + 2z^2 + z[\mathcal{S}_X(z) + \mathcal{S}_C(z) + \mathcal{K}(z)] + z^2\mathcal{S}_X'(z^2) + z^2\mathcal{S}_C'(z^2) + z[1 + \mathcal{K}'(z) + \mathcal{S}_C'(z) + \mathcal{S}_X'(z)] \\ &\quad - z[1 + \mathcal{S}_C'(z) + \mathcal{S}_X'(z)][1 + z + \mathcal{K}(z) + \mathcal{S}_C(z) + \mathcal{S}_X(z)] - z^2[1 + \mathcal{S}_X'(z^2) + \mathcal{S}_C'(z^2)] \\ &\quad + [z + \mathcal{K}(z) + \mathcal{S}_C(z)][z(1 + \mathcal{K}'(z) + \mathcal{S}_C'(z) + \mathcal{S}_X'(z)) \\ &\quad - z(1 + \mathcal{K}'(z) + \mathcal{S}_X'(z))(1 + z + \mathcal{K}(z) + \mathcal{S}_C(z) + \mathcal{S}_X(z))]. \end{aligned}$$

Evaluating this in Maple, we find that

$$z \cdot \mathcal{DH}'(z) = z + 2z^2 + 6z^3 + 24z^4 + 90z^5 + 438z^6 + 2156z^7 + 11872z^8 + 67428z^9 + 400100z^{10} + \dots,$$

so

$$\mathcal{DH}(z) = z + z^2 + 2z^3 + 6z^4 + 18z^5 + 73z^6 + 308z^7 + 1484z^8 + 7492z^9 + 40010z^{10} + \dots$$

In order to build a Boltzmann sampler for \mathcal{DH}° and hence (by Corollary 4.11) an unbiased sampler for \mathcal{DH} , we apply the rules in Table 5 and the Pólya-Boltzmann samplers described by Bodirsky *et al.* [19] to translate our symbolic specification for \mathcal{DH}° into a Boltzmann sampler. We discuss some aspects of the implementation in Section 6, and refer to the accompanying Maple code for full details.

5.5.2 Three-leaf power graphs

In this section we apply the steps outlined in Sections 4.2, 4.3, and 4.4 to enumerate and build an unbiased sampler for the class $3\mathcal{LP}$ of three-leaf power graphs. Instead of working directly with the graphs, we will again work with their split tree decompositions.

Let \mathcal{S}_X , \mathcal{S}_C , and \mathcal{A} be as in Section 5.4.2. Then we have

$$\mathcal{S}_X = \mathcal{A} \times \text{SET}_{\geq 1}(\mathcal{A} + \mathcal{S}_X)$$

$$\mathcal{S}_C = \text{SET}_{\geq 2}(\mathcal{A} + \mathcal{S}_X)$$

$$\mathcal{A} = \text{SET}_{\geq 1}(\mathcal{Z})$$

Recalling the cycle index sums for SET , SET_0 , and SET_1 from Section 5.5.1, we apply the transfer theorems from Table 4 to obtain the following equations:

$$\mathcal{S}_X(z) = \mathcal{A}(z) \cdot \left[\exp \left(\sum_{i=1}^{\infty} \frac{1}{i} (\mathcal{A}(z^i) + \mathcal{S}_X(z^i)) \right) - 1 \right]$$

$$\mathcal{S}_C(z) = \exp \left(\sum_{i=1}^{\infty} \frac{1}{i} (\mathcal{A}(z^i) + \mathcal{S}_X(z^i)) \right) - 1 - \mathcal{A}(z) - \mathcal{S}_X(z)$$

$$\mathcal{A}(z) = \exp \left(\sum_{i=1}^{\infty} \frac{1}{i} z^i \right) - 1 = \frac{z}{1-z}$$

Using `combstruct`, we can compute the coefficients of these generating functions:

$$\mathcal{S}_X(z) = z^2 + 4z^3 + 12z^4 + 36z^5 + 107z^6 + 331z^7 + 1041z^8 + 3359z^9 + 11018z^{10} + \dots$$

$$\mathcal{S}_C(z) = z^2 + 3z^3 + 11z^4 + 34z^5 + 116z^6 + 378z^7 + 1276z^8 + 4299z^9 + 14684z^{10} + \dots$$

$$\mathcal{A}(z) = z + z^2 + z^3 + z^4 + z^5 + z^6 + z^7 + z^8 + z^9 + z^{10} + \dots$$

We now build a specification for $3\mathcal{LP}^\circ$ in terms of \mathcal{S}_X , \mathcal{S}_C , and \mathcal{A} :

$$\begin{aligned} 3\mathcal{LP}^\circ &= \mathcal{Z}^\circ \\ &+ \text{SET}_2^\circ \odot \mathcal{Z} \\ &+ \mathcal{Z}^\circ \times (\mathcal{S}_X + \mathcal{S}_C + \mathcal{A} \times (\mathcal{S}_X + \mathcal{S}_C) + \text{SET}_{\geq 2}(\mathcal{Z})) \\ &+ \text{SET}_2^\circ \odot \mathcal{S}_X \\ &+ \mathcal{A} \times \text{SET}^\circ \odot (\mathcal{S}_X + \mathcal{A}) \\ &+ \text{SET}_{\geq 3}^\circ \odot \mathcal{Z} \\ &+ (\mathcal{S}_X + \mathcal{S}_C) \times \text{SET}^\circ \odot \mathcal{Z} \end{aligned}$$

Term i in this specification corresponds to case i of the following cases:

1. The tree has one leaf.
2. The tree has two leaves.
3. The marked cycle has length 1.
4. The marked cycle has length ≥ 2 and has as its center an edge connecting two star nodes at extremities.
5. The marked cycle has length ≥ 2 and has as its center a star node.
6. The marked cycle has length ≥ 2 and has as its center an isolated clique node (*i.e.* a clique node that is the entire graph).
7. The marked cycle has length ≥ 2 and has as its center a clique node connected to a star node.

Recall that in addition to the restrictions from being a distance-hereditary split tree, the split tree of a three-leaf power graph also cannot have an edge connecting the centers of two star nodes, and can only have cliques as meta-leaves connected to star nodes. Thus these seven cases cover all possibilities for a cycle-pointed split tree of a three-leaf power graph.

Next, we translate this specification into a generating function equation. Recalling the cycle index sums for $Z_{\text{SET}_2^{\mathcal{G}}}$, $Z_{\text{SET}^{\mathcal{G}}}$, and $Z_{\text{SET}_{\geq 3}^{\mathcal{G}}}$ from Section 5.5.1, we apply the transfer theorems from Table 4 to obtain an equation for the OGF of $3\mathcal{LP}^{\circ}$:

$$\begin{aligned} z \cdot 3\mathcal{LP}'(z) &= z + 2z^2 + z[\mathcal{S}_{\mathcal{X}}(z) + \mathcal{S}_{\mathcal{C}}(z) + \mathcal{A}(z)(\mathcal{S}_{\mathcal{X}}(z) + \mathcal{S}_{\mathcal{C}}(z)) + \mathcal{A}(z) - z] + z^2\mathcal{S}_{\mathcal{X}}'(z^2) \\ &\quad + \mathcal{A}(z) \left(\sum_{l=2}^{\infty} z^l (\mathcal{A}'(z^l) + \mathcal{S}_{\mathcal{X}}'(z^l)) \right) \cdot \exp \left(\sum_{i=1}^{\infty} \frac{1}{i} [\mathcal{A}(z^i) + \mathcal{S}_{\mathcal{X}}(z^i)] \right) \\ &\quad + \left(\sum_{l=2}^{\infty} z^l \right) \cdot \exp \left(\sum_{i=1}^{\infty} \frac{1}{i} z^i \right) - z^2 + [\mathcal{S}_{\mathcal{X}}(z) + \mathcal{S}_{\mathcal{C}}(z)] \left(\sum_{l=2}^{\infty} z^l \right) \cdot \exp \left(\sum_{i=1}^{\infty} \frac{1}{i} z^i \right) \end{aligned}$$

By differentiating the earlier expressions for $\mathcal{S}_{\mathcal{X}}(z)$, $\mathcal{S}_{\mathcal{C}}(z)$, and $\mathcal{A}(z)$ and using these to simplify the above formula, it follows that

$$\begin{aligned} z \cdot 3\mathcal{LP}'(z) &= z + 2z^2 + z[\mathcal{S}_{\mathcal{X}}(z) + \mathcal{S}_{\mathcal{C}}(z) + \mathcal{A}(z)(\mathcal{S}_{\mathcal{X}}(z) + \mathcal{S}_{\mathcal{C}}(z)) + \mathcal{A}(z) - z] + z^2\mathcal{S}_{\mathcal{X}}'(z^2) \\ &\quad + \mathcal{A}(z)[z(\mathcal{A}'(z) + \mathcal{S}_{\mathcal{X}}'(z) + \mathcal{S}_{\mathcal{C}}'(z)) - z(\mathcal{A}'(z) + \mathcal{S}_{\mathcal{X}}'(z))(1 + \mathcal{A}(z) + \mathcal{S}_{\mathcal{X}}(z) + \mathcal{S}_{\mathcal{C}}(z))] \\ &\quad + \frac{z^2}{(1-z)^2} - z^2 + (\mathcal{S}_{\mathcal{X}}(z) + \mathcal{S}_{\mathcal{C}}(z)) \cdot \frac{z^2}{(1-z)^2}. \end{aligned}$$

Evaluating this in Maple, we find that

$$z \cdot 3\mathcal{LP}'(z) = z + 2z^2 + 6z^3 + 20z^4 + 60z^5 + 192z^6 + 574z^7 + 1816z^8 + 5661z^9 + 18400z^{10} + \dots,$$

so

$$3\mathcal{LP}(z) = z + z^2 + 2z^3 + 5z^4 + 12z^5 + 32z^6 + 82z^7 + 227z^8 + 629z^9 + 1840z^{10} + \dots$$

In order to build a Boltzmann sampler for $3\mathcal{LP}^{\circ}$ and hence (by Corollary 4.11) an unbiased sampler for $3\mathcal{LP}$, we apply the rules in Table 5 and the Pólya-Boltzmann samplers described by Bodirsky *et al.* [19] to translate our symbolic specification for $3\mathcal{LP}^{\circ}$ into a Boltzmann sampler. We discuss some aspects of the implementation in Section 6, and refer to the accompanying Maple code for full details.

6 Implementation and empirical study of samplers

6.1 Overview

Using the computer algebra system Maple, we have implemented unbiased samplers for the class \mathcal{DH} of split trees of distance-hereditary graphs and the class $3\mathcal{LP}$ of split trees of three-leaf power graphs. These samplers take a real parameter z that is between 0 and the singularity of the generating function of the class (we discuss estimation of the singularity in Section 6.2.2), and output a string representation of the split tree that obeys the semantics in Table 6. Some examples of split tree strings and their descriptions are provided in Table 7.

In order to visualize the graphs that are being generated, we recall that the original graph of a split tree (*cf.* Definition 5.8) has one node for each leaf of the split tree, and has an edge between two nodes iff there exists a path connecting the corresponding leaves in the split tree that uses at most one internal edge of each node label. Note that, since the number of nodes of the graph is

Symbol	Meaning
Z	a leaf node
KR	a clique root (can only appear as the root of the tree)
SR	a star root (can only appear as the root of the tree)
K	a clique that has been entered from another node
SX	a star that has been entered from another node at one of its extremities
SC	a star that has been entered from another node at its center
$e(A, B)$	an edge that connects nodes A and B
$A(B_1, \dots, B_k)$	B_1, \dots, B_k are neighbors of A (if A is SR or SX, then B_1 is connected to the center of A)

Table 6. Semantics for the strings returned by our Maple implementations of the \mathcal{DH} and $3\mathcal{LP}$ samplers.

Split tree string	Description
$Z(K(Z, Z))$	a leaf connected to a clique that has two other leaves as neighbors
$KR(Z, Z, Z)$	a clique with three leaves as neighbors (same as previous)
$Z(SC(Z, Z))$	a leaf connected to the center of a star that has two leaves as its extremities
$Z(SX(Z, Z))$	a leaf connected to an extremity of a star that has a leaf as its center and a leaf as its other extremity (same as previous)
$e(SC(Z, Z), SC(Z, Z))$	an edge joining two star nodes at their centers, each of which has two leaves as its extremities
$SR(K(Z, Z), Z, Z, Z)$	a star with three leaves as its extremities and whose center is connected to a clique with two other leaves as neighbors

Table 7. Some examples of split tree strings and their descriptions.

equal to the number of leaves of the split tree, when we refer to the “size” of a split tree we mean its number of leaves.

We have built a Python package called `split_tree`, which computes the original graph corresponding to a given distance-hereditary or three-leaf power split tree string and visualizes the graph. This package has functions to translate a split tree string into an object-oriented representation of that split tree, to compute the original graph of a split tree object with the quadratic-time algorithm that checks whether each pair of vertices is or is not accessible, and to draw and export the

generated graphs using **NetworkX**. An original graph of size n is provided as a list of adjacencies over a canonical set $\{1, \dots, n\}$ of vertices, so the drawing functionality can easily be replaced with another package such as **Graphviz**.

We use this combination of Maple and Python to take advantages of the strengths of each of the languages – Maple has the powerful **comstruct** package, which allows for the automatic computation of generating function coefficients from non cycle-pointed combinatorial specifications, while Python is better suited to the object-oriented computation used in **split_tree**.

To generate graphs of large size, we employ the technique of *singular sampling*, described by Duchon *et al.* [27], in which we sample from each class at the singularity of its corresponding generating function. Figures 29, 30, and 31 depict graphs that were drawn from a singular sampler.



Figure 29. A randomly generated distance-hereditary graph with 86 vertices.

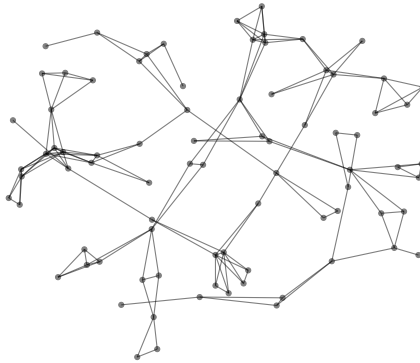


Figure 30. A randomly generated three-leaf power graph with 82 vertices.

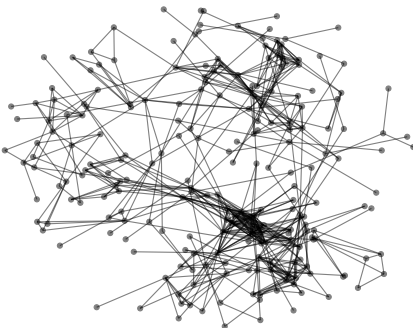


Figure 31. A randomly generated distance-hereditary graph with 224 vertices.

6.2 Details of interest

6.2.1 Oracles

For each sum of two combinatorial classes that appears in the specifications of \mathcal{DH}° and $3\mathcal{LP}^\circ$, the corresponding Boltzmann sampler has a Bernoulli switch (*cf.* Table 5) whose parameter depends on the values of the generating functions of the two classes that are being added, evaluated at the parameter z that was passed into the sampler. To compute these values, we require an *oracle* for each generating function of interest, which is a function that returns the value of the generating function at a particular input value.

Depending on the particular problem, oracles may be obtained by computing a closed form expression for the generating function, or by employing an iterative method such as Newton iteration. However since `comstruct` provides us access to the coefficients of each of the generating functions in question, we use these instead to build the necessary oracles. Specifically, for an OGF $\mathcal{A}(z)$, we build an approximate oracle by precomputing the exact values of the coefficients $\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_N$ (we take $N = 2000$ in the code) and then, on input z , returning

$$\sum_{i=0}^N \mathcal{A}_i z^i.$$

6.2.2 Radius of convergence

As mentioned above, we query our samplers at the singularity, or radius of convergence, of the corresponding generating function in order to generate graphs of large size. Finding the radius of convergence in Boltzmann sampling has traditionally been done using binary search [21, 2] on a perfect oracle (or one that can be made infinitely precise) for the generating function; however, given that we have only finitely many coefficients of the generating function, we approach this question from a slightly different angle by directly using these coefficients.³

Given the coefficients $\mathcal{A}_0, \dots, \mathcal{A}_N$ of $\mathcal{A}(z)$ (for some fixed $N > 0$), a few methods of estimating the radius of convergence $\rho_{\mathcal{A}}$ of $\mathcal{A}(z)$ immediately present themselves. We recall by the ratio and

³At the time of writing, the NewtonGF package of Pivoteau *et al.* [5] was not yet able to support cycle-pointed classes and specifications.

root convergence tests that the quantities

$$\frac{\mathcal{A}_{n-1}}{\mathcal{A}_n} \quad \text{and} \quad \frac{1}{\sqrt[n]{\mathcal{A}_n}}$$

both converge to ρ_A as $n \rightarrow \infty$ (assuming the limits exist), so we may estimate ρ_A with the values

$$\frac{\mathcal{A}_{N-1}}{\mathcal{A}_N} \quad \text{and} \quad \frac{1}{\sqrt[N]{\mathcal{A}_N}}.$$

However, we can obtain a more accurate estimate by taking into account a first-order approximation of the error between $\mathcal{A}_{N-1}/\mathcal{A}_N$ and ρ_A as a function of N . Specifically, we know from Georgescu [1] that

$$\frac{\mathcal{A}_n}{\mathcal{A}_{n-1}} \sim \frac{1}{\rho_A} \left(1 - \frac{C}{n} \right)$$

for some constant C as $n \rightarrow \infty$, so the plot of

$$\frac{\mathcal{A}_n}{\mathcal{A}_{n-1}} \quad \text{vs.} \quad \frac{1}{n}$$

is, in the limit $n \rightarrow \infty$, a straight line with slope $1/\rho_A$. We thus estimate ρ_A by making such a plot (known as a Domb-Sykes plot) over the range $N/2 \leq n \leq N$, using a linear interpolation to estimate its slope, and computing the reciprocal of this value.

To briefly compare the accuracy of these methods on a generating function whose radius of convergence is known, consider

$$\mathcal{A}(z) = \frac{z}{(1-z)^2} = \sum_{n=0}^{\infty} n z^n,$$

for which $\rho_A = 1$. Table 8 shows the estimates obtained by using the first 2000 coefficients.

Estimation method	Value	Absolute error
$\frac{\mathcal{A}_{1999}}{\mathcal{A}_{2000}}$	0.9995	$5 \cdot 10^{-4}$
$\frac{1}{\sqrt[2000]{\mathcal{A}_{2000}}}$	0.996207	$3.793 \cdot 10^{-3}$
Domb-Sykes	0.999999999966895	$3.310 \cdot 10^{-11}$

Table 8. Estimates of ρ_A for $\mathcal{A}(z) = \sum_{n=0}^{\infty} n z^n$.

Using the Domb-Sykes method, we obtain the estimates

$$\rho_{\mathcal{DH}} = 0.137935 \quad \text{and} \quad \rho_{\mathcal{LP}} = 0.259845.$$

6.2.3 Sampling of random variables

The one place where the implemented \mathcal{DH} and \mathcal{LP} samplers employ some form of rejection is in order to sample certain pairs of complicated and correlated random variables.

For example, one of the subprocedures that is used is a Boltzmann sampler $\Gamma(\text{SET}_{\geq 2}(\mathcal{A}))(z)$, where \mathcal{A} is an arbitrary class. According to the Pólya-Boltzmann sampler for $\text{SET}_{\geq 2}$ (which is a modification of the one for SET given in Figure 18), this procedure requires sampling

$$J \leftarrow \text{MAX_INDEX}_{\geq 1}(\mathcal{A}, z) \quad \text{and} \quad k_J \leftarrow \text{Pois}_{\geq 1}\left(\frac{\mathcal{A}(z^J)}{J}\right)$$

such that either $J > 1$ or $k_J > 1$ (or both). We currently do so by sampling J unconditionally using inversion sampling, then sampling k_J given the value of J (again using inversion sampling), and then checking if at least one of the values is greater than 1 and repeating if not. A similar rejection is used in the Boltzmann samplers for $\text{SET}_{\geq 2}^{\circ} \odot \mathcal{A}$ and $\text{SET}_{\geq 3}^{\circ} \odot \mathcal{A}$.

We propose that these instances of rejection can be eliminated by sampling directly from the joint distributions on the pairs of variables. Using the same example, we wish to sample from the distribution

$$\mathbb{P}[J = a, k_J = b \mid J > 1 \text{ or } k_J > 1].$$

The marginal of this distribution on J is

$$\mathbb{P}[J = a \mid J > 1 \text{ or } k_J > 1] = \begin{cases} \frac{\mathbb{P}[J = 1] - \mathbb{P}[J = 1, k_J = 1]}{1 - \mathbb{P}[J = 1, k_J = 1]} & a = 1 \\ \frac{\mathbb{P}[J = a]}{1 - \mathbb{P}[J = 1, k_J = 1]} & a > 1 \end{cases},$$

and then the conditional on k_J is

$$\mathbb{P}[k_J = b \mid J = a \text{ and } (J > 1 \text{ or } k_J > 1)] = \begin{cases} \mathbb{P}[k_1 = b \mid k_1 > 1] & a = 1 \\ \mathbb{P}[k_a = b] & a > 1 \end{cases}.$$

Since all terms on the right hand side of the above equalities except $\mathbb{P}[J = 1, k_J = 1]$ are already known from the respective individual inversion samplers on J and k_J , and $\mathbb{P}[J = 1, k_J = 1]$ can be easily computed, we may sample from the joint distribution by first sampling a from the marginal on J using inversion, and then sampling b from the conditional on k_J using inversion once again. This will correctly sample the pair (J, k_J) without using rejection.

6.3 Empirical analysis

6.3.1 Chi-squared tests

In order to analyze the accuracy of the samplers, we perform a Pearson's chi-squared test on the distribution over the possible sizes of the object generated by the sampler. Recall that for a class \mathcal{A} , the theoretical distribution of the size of an object generated by a Boltzmann sampler $\Gamma\mathcal{A}$ is

$$\mathbb{P}_z[S = n] = \frac{\mathcal{A}_n z^n}{\mathcal{A}(z)},$$

so we can compute the theoretical size distributions for our samplers $\Gamma\mathcal{DH}^{\circ}$ and $\Gamma\mathcal{ZLP}^{\circ}$ (recalling that, while the samplers are unbiased for \mathcal{DH} and \mathcal{ZLP} , they are only in fact Boltzmann samplers for \mathcal{DH}° and \mathcal{ZLP}°).

To generate empirical size distributions, we sample $N = 1000$ objects from each class with parameter $z = 0.1$, and maintain bucket counts O_i for $1 \leq i \leq n - 1$ ($n = 30$) of the number of trees

of size i that have been drawn, and a count O_n of the number of trees of size at least n that have been drawn. Then, the χ^2 statistic is

$$\chi^2 = \sum_{i=1}^n \frac{(O_i - Np_i)^2}{Np_i}$$

where p_i is the theoretical probability of being in bucket i [16].

Running this procedure for each class, we obtain values of

$$\chi_{\mathcal{DH}^\odot}^2 = 15.49 \quad \text{and} \quad \chi_{\mathcal{3LP}^\odot}^2 = 8.146.$$

Since the test has $n - 1$ degrees of freedom, the cutoff statistic value for $p = 0.05$ is 43.7 [25]. As both of the computed statistics are (well) below this value, in both cases we fail to reject the null hypothesis that the sampler produces the correct size distribution.

Figures 32 and 33 provide a graphical depiction of the agreement between the theoretical and empirical size distributions.

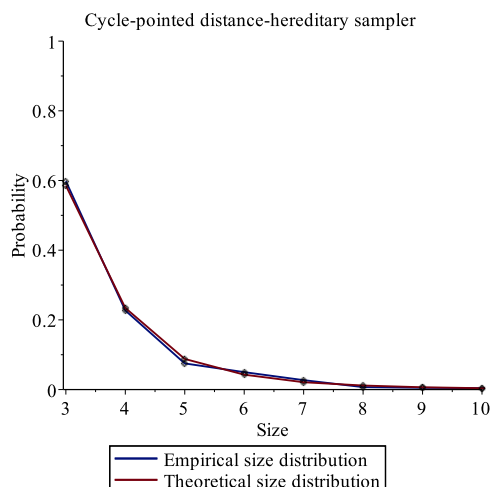


Figure 32. Theoretical size distribution, & empirical distribution from 1000 samples of $\Gamma\mathcal{DH}^\odot(0.1)$.

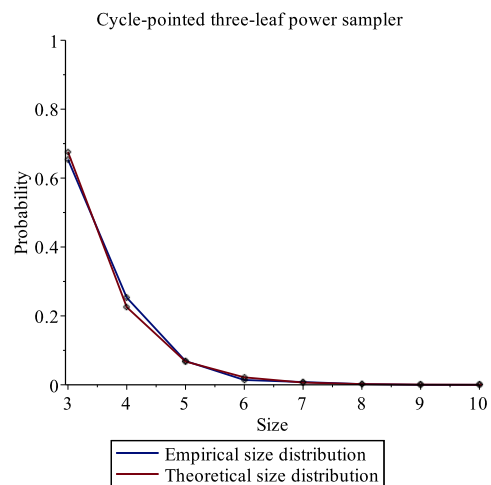


Figure 33. Theoretical size distribution, & empirical distribution from 1000 samples of $\Gamma\mathcal{3LP}^\odot(0.1)$.

6.3.2 Timing studies

We recall that standard Boltzmann samplers run in linear time in the size of the object that is returned. In order to study the running time of our cycle-pointed samplers, we repeatedly sample from them at a fixed parameter value, and for each sample we measure the size of the returned tree (*i.e.* the number of leaves it has) and the time it took to generate.

Figures 34 and 35 show scatter plots of time vs. size for 2000 graphs drawn from each Boltzmann sampler, with lines of best fit included. Both graphs show a distinct linear relationship between the two variables.

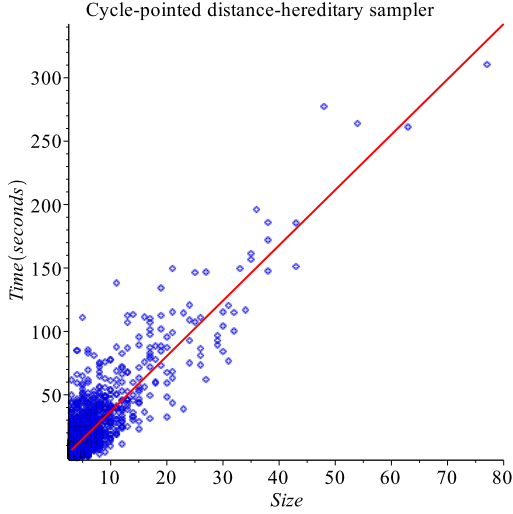


Figure 34. Scatter plot of time vs. size for 2000 samples of $\Gamma\mathcal{DH}^\circ(0.13)$, with line of best fit.

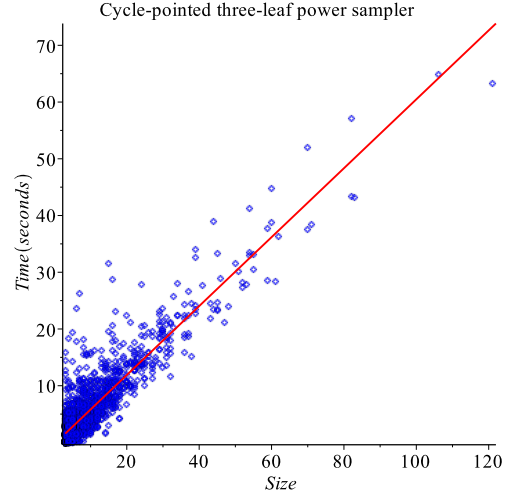


Figure 35. Scatter plot of time vs. size for 2000 samples of $\Gamma\mathcal{3LP}^\circ(0.25)$, with line of best fit.

7 Conclusion

In this work, we study the problem of enumerating and sampling from combinatorial classes of unlabeled and unrooted graphs. We consider two techniques for addressing this problem: the dissymmetry theorem, which allows for the analysis of an unrooted class of trees by first analyzing three rooted counterparts; and cycle pointing, which marks certain cycles of a graph in such a way that there are exactly n distinct pointed graphs for each unpointed graph of size n , thereby allowing for a straightforward translation of the analysis of the pointed class into an analysis of the unpointed class.

While at first glance the dissymmetry theorem provides only the enumeration of the unrooted class, we have shown how to build a Boltzmann sampler for an arbitrary combinatorial class specified by the dissymmetry theorem, assuming that there exist samplers for the corresponding vertex-rooted and undirected edge-rooted classes. Secondly, we have provided an exposition of cycle pointing that focuses on the enumeration and unbiased sampling of the underlying unpointed class, in the hope of elucidating this technique for future practitioners. Finally, we have applied the technique of cycle pointing to build the first unbiased samplers for the classes of distance-hereditary graphs and three-leaf power graphs.

Much further work remains. As a small point, we will make the changes to the implementation described in Section 6.2.3 so that the cycle-pointed samplers no longer use any form of rejection. Also, our distance-hereditary and three-leaf power samplers provide a fertile starting point for analyzing parameters of these graphs, whether parameters of the split trees or of the graphs themselves. For example, in Figures 36 and 37 we draw from $\Gamma\mathcal{DH}^\circ$ to estimate the average number of clique nodes and star nodes in a random distance-hereditary split tree of size n as a function of n . We see that both of these parameters appear to grow linearly with n , and we conjecture that the number of clique nodes grows as approximately $\sim 0.221n$ and the number of star nodes grows as approximately $\sim 0.593n$. One interesting suggestion is that an understanding of the distribution of parameters in random graphs from these classes can be applied, for example, to see if the phylogenetic trees studied by Nishimura *et al.* [24] appear to behave randomly. Finally, we propose that the techniques

used in this work can be applied to many other classes of unlabeled and unrooted graphs which have not yet been analyzed in this manner.

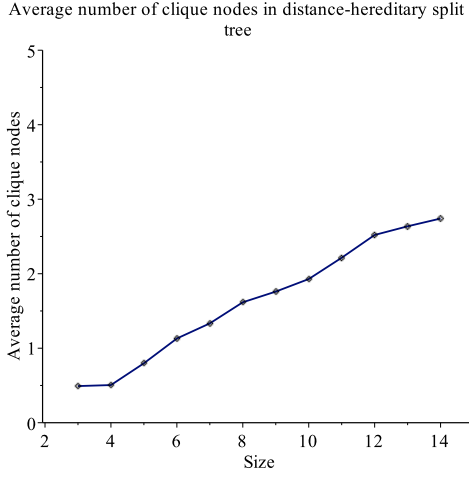


Figure 36. Average number of clique nodes in a distance-hereditary split tree of size n vs. n , from 2000 samples of $\Gamma\mathcal{DH}^\circ(0.137)$.

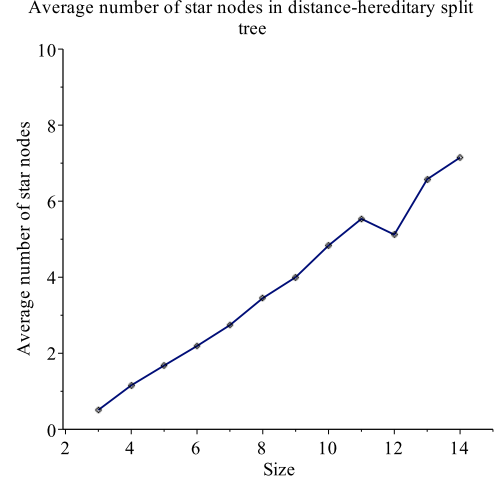


Figure 37. Average number of star nodes in a distance-hereditary split tree of size n vs. n , from 2000 samples of $\Gamma\mathcal{DH}^\circ(0.137)$.

Appendices

A Commonly used notation

Notation	Meaning
\mathcal{A}^\bullet	Vertex-rooted class of \mathcal{A}
$\mathcal{A}^{\bullet-\bullet}$	Undirected edge-rooted class of \mathcal{A}
$\mathcal{A}^{\bullet\rightarrow\bullet}$	Directed edge-rooted class of \mathcal{A}
$\text{Sym}(G)$	Symmetries of G
$\text{RSym}((G, c))$	Rooted c -symmetries of (G, c)
\mathcal{A}°	Cycle-pointed class of \mathcal{A}
$\mathcal{A}^\mathfrak{S}$	Symmetric cycle-pointed class of \mathcal{A}
$\mathcal{A} \odot \mathcal{B}$	Cycle-pointed substitution
$Z_{\mathcal{A}}$	Cycle index sum of \mathcal{A}
$\Gamma\mathcal{A}$	Boltzmann sampler for \mathcal{A}
$\Gamma\mathbb{Z}_{\mathcal{A}}$	Pólya-Boltzmann sampler for \mathcal{A}

Table 9. Commonly used notation in this work.

B Unbiased sampler for the class of unlabeled, unrooted, non-plane 2-3 trees

We recall from Section 4.2.2 the decomposition of the class \mathcal{T}° of unlabeled, unrooted, non-plane trees:

$$\mathcal{T}^\circ = \mathcal{Z}^\bullet \times \text{SET}_{1,3,4}(\mathcal{S}) + \mathcal{Z} \times (\text{SET}_{3,4}^{\mathcal{S}} \odot \mathcal{S}) + \text{SET}_2^{\mathcal{S}} \odot \mathcal{S}$$

$$\mathcal{S} = \mathcal{Z} + \mathcal{Z} \times \text{SET}_{2,3}(\mathcal{S}).$$

To build Boltzmann samplers for these classes, we require Boltzmann samplers for $\text{SET}_k(\mathcal{A})$, $\text{SET}_k^\circ \odot \mathcal{A}$, and $\text{SET}_k^{\mathcal{S}} \odot \mathcal{A}$ (for arbitrary k and \mathcal{A}), which are shown in Figures 38, 39, and 40.⁴

```

Γ(SETk(A))(z):
  (n1, ..., nk) ← PARTITION(A, k, z)
  S ← null
  for i from 1 to k do
    for j from 1 to ni do
      γ ← ΓA(zi)
      Add i copies of γ to S
    end for
  end for
  return S

```

Figure 38. Boltzmann sampler for $\text{SET}_k(\mathcal{A})$.

⁴PARTITION(\mathcal{A}, k, z) is a random generator over the tuples (n_1, \dots, n_k) of non-negative integers such that $\sum_{i=1}^k in_i = k$, for the distribution

$$\mathbb{P}[(n_1, \dots, n_k)] = \frac{(\mathcal{A}(z))^{n_1} (\mathcal{A}(z^2))^{n_2} \dots (\mathcal{A}(z^k))^{n_k} \cdot [s_1^{n_1} s_2^{n_2} \dots s_k^{n_k}] Z_{\text{SET}_k}(s_1, s_2, \dots)}{Z_{\text{SET}_k}(\mathcal{A}(z), \mathcal{A}(z^2), \dots)}.$$

MARKED_PARTITION(\mathcal{A}, k, z) is a random generator over the tuples (ℓ, n_1, \dots, n_k) of non-negative integers such that $\ell + \sum_{i=1}^k in_i = k$ and $\ell \geq 1$, for the distribution

$$\mathbb{P}[(\ell, n_1, \dots, n_k)] = \frac{z^\ell \mathcal{A}'(z^\ell) (\mathcal{A}(z))^{n_1} (\mathcal{A}(z^2))^{n_2} \dots (\mathcal{A}(z^k))^{n_k} \cdot [t_\ell s_1^{n_1} s_2^{n_2} \dots s_k^{n_k}] Z_{\text{SET}_k^\circ}(s_1, s_2, \dots; t_1, t_2, \dots)}{Z_{\text{SET}_k^\circ}(\mathcal{A}(z), \mathcal{A}(z^2), \dots; z\mathcal{A}'(z), z^2\mathcal{A}'(z^2), \dots)}.$$

MARKED_SYMM_PARTITION(\mathcal{A}, k, z) is a random generator over the tuples (ℓ, n_1, \dots, n_k) of non-negative integers such that $\ell + \sum_{i=1}^k in_i = k$ and $\ell \geq 2$, for the distribution

$$\mathbb{P}[(\ell, n_1, \dots, n_k)] = \frac{z^\ell \mathcal{A}'(z^\ell) (\mathcal{A}(z))^{n_1} (\mathcal{A}(z^2))^{n_2} \dots (\mathcal{A}(z^k))^{n_k} \cdot [t_\ell s_1^{n_1} s_2^{n_2} \dots s_k^{n_k}] Z_{\text{SET}_k^{\mathcal{S}}}(s_1, s_2, \dots; t_1, t_2, \dots)}{Z_{\text{SET}_k^{\mathcal{S}}}(\mathcal{A}(z), \mathcal{A}(z^2), \dots; z\mathcal{A}'(z), z^2\mathcal{A}'(z^2), \dots)}.$$

```

 $\Gamma(\text{SET}_k^{\odot} \odot \mathcal{A})(z)$ :
   $(\ell, n_1, \dots, n_k) \leftarrow \text{MARKED\_PARTITION}(\mathcal{A}, k, z)$ 
   $S \leftarrow \text{null}$ 
  for  $i$  from 1 to  $k$  do
    for  $j$  from 1 to  $n_i$  do
       $\gamma \leftarrow \Gamma\mathcal{A}(z^i)$ 
      Add  $i$  copies of  $\gamma$  to  $S$ 
    end for
  end for
   $(\gamma, c) \leftarrow \Gamma\mathcal{A}^{\odot}(z^{\ell})$ 
  Add  $\ell$  copies of  $(\gamma, c)$  to  $S$ 
  Let  $c'$  be the cycle obtained by composing the  $\ell$  copies of  $c$ 
  return  $(S, c)$ 

```

Figure 39. Boltzmann sampler for $\text{SET}_k^{\odot} \odot \mathcal{A}$.

```

 $\Gamma(\text{SET}_k^{\otimes} \odot \mathcal{A})(z)$ :
   $(\ell, n_1, \dots, n_k) \leftarrow \text{MARKED\_SYMM\_PARTITION}(\mathcal{A}, k, z)$ 
   $S \leftarrow \text{null}$ 
  for  $i$  from 1 to  $k$  do
    for  $j$  from 1 to  $n_i$  do
       $\gamma \leftarrow \Gamma\mathcal{A}(z^i)$ 
      Add  $i$  copies of  $\gamma$  to  $S$ 
    end for
  end for
   $(\gamma, c) \leftarrow \Gamma\mathcal{A}^{\odot}(z^{\ell})$ 
  Add  $\ell$  copies of  $(\gamma, c)$  to  $S$ 
  Let  $c'$  be the cycle obtained by composing the  $\ell$  copies of  $c$ 
  return  $(S, c)$ 

```

Figure 40. Boltzmann sampler for $\text{SET}_k^{\otimes} \odot \mathcal{A}$.

We then apply these rules to build Boltzmann samplers for \mathcal{S} , \mathcal{S}° , and \mathcal{T}° , which are shown in Figures 41, 42, and 43. By Corollary 4.11, running $\Gamma\mathcal{T}^\circ(z)$ and forgetting the marked cycle provides an unbiased sampler for the class \mathcal{T} of unlabeled, unrooted, non-plane 2-3 trees.

We use Z to denote a node, Z^\bullet to denote a node with a marked singleton cycle, $A = (B_1, \dots, B_k)$ to denote that B_1, \dots, B_k are neighbors of A , and $e(A, B)$ to denote that A and B are connected by an edge. Also for the sake of simplicity, we let

$$\text{DRAW}(\alpha_1, \dots, \alpha_k)$$

be a random generator that draws each integer $1 \leq i \leq k$ with probability

$$\frac{\alpha_i}{\alpha_1 + \dots + \alpha_k}.$$

```

 $\Gamma\mathcal{S}(z)$ :
   $i \leftarrow \text{DRAW} \left( z, \frac{1}{2}z\mathcal{S}(z)^2, \frac{1}{2}z\mathcal{S}(z^2), \frac{1}{6}z\mathcal{S}(z)^3, \frac{1}{2}z\mathcal{S}(z)\mathcal{S}(z^2), \frac{1}{3}z\mathcal{S}(z^3) \right)$ 
  switch  $i$ 
    case 1
      return  $Z$ 
    case 2
      return  $Z(\Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z))$ 
    case 3
       $\gamma \leftarrow \Gamma\mathcal{S}(z^2)$ 
      return  $Z(\gamma, \gamma)$ 
    case 4
      return  $Z(\Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z))$ 
    case 5
       $\gamma \leftarrow \Gamma\mathcal{S}(z^2)$ 
      return  $Z(\Gamma\mathcal{S}(z), \gamma, \gamma)$ 
    case 6
       $\gamma \leftarrow \Gamma\mathcal{S}(z^3)$ 
      return  $Z(\gamma, \gamma, \gamma)$ 

```

Figure 41. Boltzmann sampler for \mathcal{S} .

```

 $\Gamma\mathcal{S}^\odot(z)$ :
   $i \leftarrow \text{DRAW} \left( z, \frac{1}{2}z\mathcal{S}(z)^2, \frac{1}{2}z\mathcal{S}(z^2), \frac{1}{6}z\mathcal{S}(z)^3, \frac{1}{2}z\mathcal{S}(z)\mathcal{S}(z^2), \frac{1}{3}z\mathcal{S}(z^3), z^2\mathcal{S}'(z)\mathcal{S}(z), \right.$ 
     $\left. z^3\mathcal{S}'(z^2), \frac{1}{2}z^2\mathcal{S}'(z)\mathcal{S}(z)^2, \frac{1}{2}z^2\mathcal{S}'(z)\mathcal{S}(z^2), z^3\mathcal{S}'(z^2)\mathcal{S}(z), z^4\mathcal{S}'(z^3) \right)$ 
  switch  $i$ 
  case 1
    return  $Z^\bullet$ 
  case 2
    return  $Z^\bullet(\Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z))$ 
  case 3
     $\gamma \leftarrow \Gamma\mathcal{S}(z^2)$ 
    return  $Z^\bullet(\gamma, \gamma)$ 
  case 4
    return  $Z^\bullet(\Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z))$ 
  case 5
     $\gamma \leftarrow \Gamma\mathcal{S}(z^2)$ 
    return  $Z^\bullet(\Gamma\mathcal{S}(z), \gamma, \gamma)$ 
  case 6
     $\gamma \leftarrow \Gamma\mathcal{S}(z^3)$ 
    return  $Z^\bullet(\gamma, \gamma, \gamma)$ 
  case 7
    return  $Z(\Gamma\mathcal{S}^\odot(z), \Gamma\mathcal{S}(z))$ 
  case 8
     $\gamma \leftarrow \Gamma\mathcal{S}^\odot(z^2)$ 
    return  $Z(\gamma, \gamma)$ , with the cycles on the two copies of  $\gamma$  composed
  case 9
    return  $Z(\Gamma\mathcal{S}^\odot(z), \Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z))$ 
  case 10
     $\gamma \leftarrow \Gamma\mathcal{S}(z^2)$ 
    return  $Z(\Gamma\mathcal{S}^\odot(z), \gamma, \gamma)$ 
  case 11
     $\gamma \leftarrow \Gamma\mathcal{S}^\odot(z^2)$ 
    return  $Z(\gamma, \gamma, \Gamma\mathcal{S}(z))$ , with the cycles on the two copies of  $\gamma$  composed
  case 12
     $\gamma \leftarrow \Gamma\mathcal{S}^\odot(z^3)$ 
    return  $Z(\gamma, \gamma, \gamma)$ , with the cycles on the three copies of  $\gamma$  composed

```

Figure 42. Boltzmann sampler for \mathcal{S}^\odot .

$\Gamma\mathcal{T}^\circ(z)$:

$i \leftarrow \text{DRAW} \left(z\mathcal{S}(z), \frac{1}{6}z\mathcal{S}(z)^3, \frac{1}{2}z\mathcal{S}(z)\mathcal{S}(z^2), \frac{1}{3}z\mathcal{S}(z^3), \frac{1}{24}z\mathcal{S}(z)^4, \frac{1}{4}z\mathcal{S}(z)^2\mathcal{S}(z^2), \frac{1}{8}z\mathcal{S}(z^2)^2, \right.$
 $\frac{1}{3}z\mathcal{S}(z)\mathcal{S}(z^3), \frac{1}{4}z\mathcal{S}(z^4), z^3\mathcal{S}'(z^2)\mathcal{S}(z), z^4\mathcal{S}'(z^3), \frac{1}{2}z^3\mathcal{S}'(z^2)\mathcal{S}(z)^2,$
 $\left. \frac{1}{2}z^3\mathcal{S}'(z^2)\mathcal{S}(z^2), z^4\mathcal{S}'(z^3)\mathcal{S}(z), z^5\mathcal{S}'(z^4), z^2\mathcal{S}'(z^2) \right)$

switch i

case 1

return $Z^\bullet(\Gamma\mathcal{S}(z))$

case 2

return $Z^\bullet(\Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z))$

case 3

$\gamma \leftarrow \Gamma\mathcal{S}(z^2)$

return $Z^\bullet(\Gamma\mathcal{S}(z), \gamma, \gamma)$

case 4

$\gamma \leftarrow \Gamma\mathcal{S}(z^3)$

return $Z^\bullet(\gamma, \gamma, \gamma)$

case 5

return $Z^\bullet(\Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z))$

case 6

$\gamma \leftarrow \Gamma\mathcal{S}(z^2)$

return $Z^\bullet(\Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z), \gamma, \gamma)$

case 7

$\gamma \leftarrow \Gamma\mathcal{S}(z^2)$

$\zeta \leftarrow \Gamma\mathcal{S}(z^2)$

return $Z^\bullet(\gamma, \gamma, \zeta, \zeta)$

case 8

$\gamma \leftarrow \Gamma\mathcal{S}(z^3)$

return $Z^\bullet(\Gamma\mathcal{S}(z), \gamma, \gamma, \gamma)$

case 9

$\gamma \leftarrow \Gamma\mathcal{S}(z^4)$

return $Z^\bullet(\gamma, \gamma, \gamma, \gamma)$

case 10

$\gamma \leftarrow \Gamma\mathcal{S}^\circ(z^2)$

return $Z(\gamma, \gamma, \Gamma\mathcal{S}(z))$, with the cycles on the two copies of γ composed

case 11

$\gamma \leftarrow \Gamma\mathcal{S}^\circ(z^3)$

return $Z(\gamma, \gamma, \gamma)$, with the cycles on the three copies of γ composed

case 12

$\gamma \leftarrow \Gamma\mathcal{S}^\circ(z^2)$

return $Z(\gamma, \gamma, \Gamma\mathcal{S}(z), \Gamma\mathcal{S}(z))$, with the cycles on the two copies of γ composed

case 13

$\gamma \leftarrow \Gamma\mathcal{S}^\circ(z^2)$

$\zeta \leftarrow \Gamma\mathcal{S}(z^2)$

return $Z(\gamma, \gamma, \zeta, \zeta)$, with the cycles on the two copies of γ composed

case 14

$\gamma \leftarrow \Gamma\mathcal{S}^\circ(z^3)$

return $Z(\gamma, \gamma, \gamma, \Gamma\mathcal{S}(z))$, with the cycles on the three copies of γ composed

case 15

$\gamma \leftarrow \Gamma\mathcal{S}^\circ(z^4)$

return $Z(\gamma, \gamma, \gamma, \gamma)$, with the cycles on the four copies of γ composed

case 16

$\gamma \leftarrow \Gamma\mathcal{S}^\circ(z^2)$

return $e(\gamma, \gamma)$, with the cycles on the two copies of γ composed

Figure 43. Boltzmann sampler for \mathcal{T}° .

References

- [1] Adelina Georgescu. *Asymptotic Treatment of Differential Equations*. Chapman & Hall, 1995.
- [2] Alexis Darrasse. *Complex tree-like structures: combinatorial analysis, random sampling and applications*. PhD thesis, Université Pierre-et-Marie-Curie, 2010.
- [3] Alexis Darrasse and Michèle Soria. Degree distribution of random Apollonian network structures and Boltzmann sampling. *DMTCS Proceedings, 2007 Conference on Analysis of Algorithms (AofA '07)*, 2008.
- [4] Alexis Darrasse, Konstantinos Panagiotou, Olivier Roussel, and Michèle Soria. Biased Boltzmann samplers and generation of extended linear languages with shuffle. *DMTCS Proceedings*, 1:125–140, 2012.
- [5] Carine Pivoteau, Pablo Rotondo, and Bruno Salvy. The NewtonGF package. <http://perso.ens-lyon.fr/bruno.salvy/software/the-newtongf-package/>.
- [6] Cedric Chauve, Éric Fusy, and Jérémie Lumbroso. An Enumeration of Distance-Hereditary and 3-Leaf Power Graphs. Preprint presented at ICGT 2014, 2014.
- [7] Edward Howorka. A characterization of distance-hereditary graphs. *The Quarterly Journal of Mathematics*, 28:417–420, 1977.
- [8] Emeric Gioan and Christophe Paul. Split Decomposition and Graph-Labelled Trees: Characterizations and Fully-Dynamic Algorithms for Totally Decomposable Graphs. *Discrete Applied Mathematics*, 160:708–733, 2012.
- [9] Éric Fusy. Uniform Random Sampling of Planar Graphs in Linear Time. *Random Structures & Algorithms*, 35:464–522, 2009.
- [10] François Bergeron, Gilbert Labelle, and Pierre Leroux. *Combinatorial Species and Structures*. Cambridge University Press, 1997.
- [11] George Pólya. *Kombinatorische Anzahlbestimmungen für Gruppen, Graphen, und chemische Verbindungen*. *Acta Mathematica*, 68:145–254, 1937.
- [12] Guillaume Chapuy, Éric Fusy, Mihyun Kang, and Bilyana Shoilekova. A Complete Grammar for Decomposing a Family of Graphs into 3-connected Components. *The Electronic Journal of Combinatorics*, 15, 2008.
- [13] H. N. de Ridder *et al.* Graphclass: distance-hereditary. http://www.graphclasses.org/classes/gc_80.html.
- [14] H. N. de Ridder *et al.* Graphclass: k -leaf power, fixed k . http://www.graphclasses.org/classes/gc_649.html.
- [15] Haiko Müller and Falk Nicolai. Polynomial time algorithms for Hamiltonian problems on bipartite distance-hereditary graphs. *Information Processing Letters*, 46:225–230, 1993.
- [16] John Walker. Chi-Square Calculator. <http://bavard.fourmilab.ch/rpkp/experiments/analysis/chiCalc.html>.
- [17] Luc Devroye. *Non-uniform Random Variate Generation*. Springer-Verlag, 1986.
- [18] Manuel Bodirsky, Éric Fusy, Mihyun Kang, and Stefan Vigerske. An unbiased pointing operator for unlabeled structures, with applications to counting and sampling. *SODA '07 Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 356–365, 2007.

- [19] Manuel Bodirsky, Éric Fusy, Mihyun Kang, and Stefan Vigerske. Boltzmann Samplers, Pólya Theory, and Cycle Pointing. <http://arxiv.org/pdf/1003.4546v2.pdf>, 2011.
- [20] Michael Artin. *Algebra*. Pearson, 2010.
- [21] Michèle Soria, Bruno Salvy, and Carine Pivoteau. Algorithms for combinatorial structures: Well-founded systems and Newton iterations. *Journal of Combinatorial Theory*, 119:1711–1773, 2012.
- [22] Miklos Bóna. *Handbook of Enumerative Combinatorics*. Taylor & Francis Group, LLC, 2015.
- [23] Mireille Bousquet-Mélou and Kerstin Weller. Asymptotic properties of some minor-closed classes of graphs. *Combinatorics, Probability and Computing*, 23(5):749–795, 2014.
- [24] Naomi Nishimura and Prabhakar Ragde. On Graph Powers for Leaf-Labeled Trees. *Journal of Algorithms*, 42:69–108, 2002.
- [25] NIST. Critical Values of the Chi-Square Distribution. <http://www.itl.nist.gov/div898/handbook/eda/section3/eda3674.htm>.
- [26] Olivier Cogis and Eric Thierry. Computing maximum stable sets for distance-hereditary graphs. *Discrete Optimization*, 2:185–188, 2005.
- [27] Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Boltzmann Samplers for the Random Generation of Combinatorial Structures. *Combinatorics, Probability and Computing*, 13:577–625, 2004.
- [28] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.
- [29] Philippe Flajolet, Éric Fusy, and Carine Pivoteau. Boltzmann Sampling of Unlabelled Structures. *SIAM Proceedings in Applied Mathematics*, 126:201–211, 2007.
- [30] Philippe Flajolet, Paul Zimmermann, and Bernard Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132:1–35, 1993.
- [31] Richard Otter. The number of trees. *Annals of Mathematics*, 49:583–599, 1948.
- [32] Sun-yuan Hsieh, Chin-wen Ho, Tsan-sheng Hsu, and Ming-tat Ko. Efficient algorithms for the Hamiltonian problem on distance-hereditary graphs. *Computing and Combinatorics*, 2387:51–75, 2002.